AD/A-002 856

# A STRUCTURAL WEIGHT ESTIMATION PROGRAM (SWEEP) FOR AIRCRAFT. VOLUME IV - MATERIAL PROPERTIES, STRUCTURE TEMPERATURE, FLUTTER AND FATIGUE

H. Haroldson, et al

Rockwell International Corporation

Prepared for:

Aeronautical Systems Division

June 1974

Best
Available
Copy

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered) AD/A005 556

# REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: ASD/XR 74-10
2. GOVT ACCESSION NO.
3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle): A Structural Weight Estimation Program (SWEEP) for Aircraft. Volume IV, Material Properties, Structure Temperature, Flutter, and Fatigue.
5. TYPE OF REPORT & PERIOD COVERED
6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s): H. Haroldson, C. Hodson, S. Mellin, H. Rockwell, S. Tejani, C. Martindale
8. CONTRACT OR GRANT NUMBER(s): F33615-71-C-1922
9. PERFORMING ORGANIZATION NAME AND ADDRESS: Rockwell International Corp, L.A. Aircraft Div, Los Angeles International Airport, Los Angeles, California 90009
10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: FX2826-71-01876/C093
11. CONTROLLING OFFICE NAME AND ADDRESS: Deputy for Development Planning, Air Force Systems Command, Wright-Patterson Air Force Base, Ohio
12. REPORT DATE: June 1974
13. NUMBER OF PAGES: ~~405~~ 408
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)
15. SECURITY CLASS. (of this report): Unclassified
15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
16. DISTRIBUTION STATEMENT (of this Report): Approved for public release; distribution unlimited.
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)



18. SUPPLEMENTARY NOTES



19. KEY WORDS (Continue on reverse side if necessary and identify by block number): weight estimation, structural weights, integrated computer programs, preliminary weight estimation, first-order weight estimations, aircraft structure weights, aircraft structural weight optimization, flutter optimization program, structural synthesis

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

Three computer programs were written with the objective of predicting the structural weight of aircraft through analytical methods. The first program, the structural weight estimation program (SWEEP), is a completely integrated program including routines for airloads, loads spectra, skin temperatures, material properties, flutter stiffness requirements, fatigue life, structural sizing, and for weight estimation of each of the major aircraft structural components. The program produces first-order weight estimates

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

(i)

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

and indicates trends when parameters are varied. Fighters, bombers, and cargo aircraft can be analyzed by the program. The program operates within 100,000 octal units on the Control Data Corporation 6600 computer. Two stand-alone programs operating within 100,000 octal units were also developed to provide optional data sources for SWEEP. These include (1) the flexible airloads program to assess the effects of flexibility on lifting surface airloads, and (2) the flutter optimization program to optimize the stiffness distribution required for lifting surface flutter prevention.

The final report is composed of 11 volumes. This volume (volume IV) contains the methodology, program description, and user's information for the use of material properties, flutter and temperature module, and fatigue module of SWEEP.

A

# NOTICES

When Government drawings, specifications, or other data are used for any purpose other than in connection with a definitely related Government procurement operation, the United States Government thereby incurs no responsibility nor any obligation whatsoever; and the fact that the Government may have formulated, furnished or in any way supplied the said drawings, specifications, or other data is not to be regarded by implication or otherwise as in any manner licensing the holder or any other person or corporation, or conveying any rights or permission to manufacture, use, or sell any patented invention that may in any way be related thereto.

Publication of this technical report does not constitute Air Force approval of the report's findings or conclusions. It is published only for the exchange and stimulation of ideas.

James H. Hall

JAMES H. HALL, Colonel, USAF
Deputy for Development Planning

# PREFACE

This report was prepared by Rockwell International Corporation, Los Angeles Aircraft Division, Los Angeles, California, under Contract F33615-71-C-1922, No. FX2826-71-01876/C093. The work was performed for the Deputy for Development Planning, Air Force System Command, Wright-Patterson Air Force Base, Ohio, and extended from September 1971 to June 1974.

Eugene L. Bahns, ASD/XRHD, was the Air Force program manager. Leonard Ascani was the program manager for Rockwell International. Other Rockwell personnel contributing to the project included:

G. Hayase - Mass Properties
R. Hiyama - Mass Properties
D. Chaloff - Mass Properties
C. Martindale - Mass Properties
H. Rockwell - Mass Properties
R. Allen - Mass Properties
P. Wildermuth - Airloads
G. Rothamer - Airloads
T. Byar - Airloads
S. Siegel - Structural Dynamics
S. Mellin - Structure and Fatigue
H. Haroldson - Thermodynamics
D. Konishi - Advanced Composites
C. Hodson - Structural Dynamics

The final report was published in 11 volumes; the complete list is as follows:

| Volume | |
|---|---|
| I | "Executive Summary" |
| II | "Program Integration and Data Management Module" |
| III | "Airloads Estimation Module" |
| IV | "Material Properties, Structure Temperature, Flutter, and Fatigue" |
| V | "Air Induction System and Landing Gear Modules" |
| VI | "Wing and Empennage Module" |
| VII | "Fuselage Module" |
| VIII | "Programmer's Manual" |
| IX | "User's Manual" |
| X | "Flutter Optimization Stand-Alone Program" |
| XI | "Flexible Airloads Stand-Alone Program" |

TABLE OF CONTENTS



**Preceding page blank**

| Section | | Page |
|---|---|---|
| | <u>Part 2 - Flutter and Temperature Module</u> | 43 |
| I | INTRODUCTION AND SUMMARY | 44 |
| | Program Objectives | 44 |
| | Module Structure and Operation | 44 |
| | Module Input and Output | 44 |
| | Approach to Temperature Evaluation | 45 |
| | The Nature of Thermal Predictions in Preliminary Design | 45 |
| | The Use of Skin Temperatures | 45 |
| | Limitations and Accuracy | 46 |
| | Approach to Flutter Estimation | 49 |
| | Background | 49 |
| | Scope | 50 |
| II | METHODOLOGY AND FORMULATIONS | 52 |
| | Temperature Methodology | 52 |
| | Heat Balance | 52 |
| | Flux Equation | 53 |
| | Solar Heat Flux | 54 |
| | Solar Flux | 54 |
| | Absorptivity and Emissivity | 54 |
| | Radiation Flux Between Skin and Ambient | 56 |
| | Nocturnal Radiation | 56 |
| | Aerodynamic Heat Flux | 57 |
| | Atmospheric Properties | 57 |
| | Local Static Pressure | 57 |
| | Local Static Temperature | 58 |

Section

Page

Section Page

## LIST OF ILLUSTRATIONS

| Figure | Title | Page |
|---|---|---|

Part 3 - Fatigue Module

LIST OF TABLES

## INTRODUCTION TO VOLUME IV

The Structural Weight Estimation Program (SWEEP) was developed as an aircraft structural prediction tool suitable for use in the preliminary design phase of vehicle synthesis. To insure prediction of realistic trend data, it considers many problem- and component-dependent criteria, tasks which are normally performed by different engineering disciplines. The functions of data development and assessment have been integrated into various program modules so that criteria and environment considerations are consistent. The purpose of this volume is to present methods and formulations and to discuss program routines that evaluate material properties, structure temperature, flutter, and fatigue.

1. Part 1 describes the material library files, data arrangement, development of stress-strain curves, and routines within the different program modules which process data from these files.

2. Part 2 describes the methodology and program of the flutter and temperature module. This module performs two basic tasks:

   a. Calculates structural temperatures at the various speeds and altitudes where loading and flutter analyses are performed.

   b. Determines the mach-altitude points most critical to the flutter evaluation of each lifting surface, and sets up the various flutter-related parameters and material properties to be used in the wing and empennage module flutter analyses. The module establishes these parameters for the following lifting surface configurations:

      (1) Wing

         (a) Swept
         (b) Unswept

      (2) Horizontal tail

      (3) Vertical tail

         (a) Conventional arrangement
         (b) T-tail arrangement

   Descriptions, autoflow charts, and program listings, of routines used to accomplish these tasks are included within Part 2.

5. Part 3 describes the methodology and program of the fatigue module. This module evaluates wing bending moment and cabin pressurization spectra, and determines the nominal spectrum stresses required to achieve the specified vehicle life. A correlation between design loads and spectrum loads is then used to establish allowable not-to-be-exceeded fatigue cutoff stresses. These allowables are used by the structural synthesis modules in sizing the structure. Wing spectrum loads are either user input data or are calculated by the airloads module. Derivation of these loads is presented in Volume III.

Descriptions, autoflow charts, and program listings of the routines used by the fatigue module are included in Part 3.

# PART 1

# MATERIAL PROPERTIES

## Section I

## INTRODUCTION

The structural synthesis procedures of an analytical weight-estimating program are not only dependent upon the modeling of general structural geometries, design loads, and requirements, but also on the modeling of the physical and mechanical properties of the materials selected for the analysis. Material descriptions must be in a form that can be used to reflect their behavior under load so that structures can be synthesized to satisfy conditions of strength, stiffness, and stability. SWEEP includes a procedure to describe the physical and mechanical properties of materials selected for analysis of each air vehicle component. This is accomplished with a material library and special routines designed to provide required material descriptions for each weight estimating module. Updating and expansion of the data bank require only the addition of data sets describing the materials to be changed or added.

The material properties library consists of a maximum of 20 sets of material data stored, at run time, on mass storage files. This file includes a full range of physical and mechanical properties for structural materials such as aluminum, titanium, and steel. The various alloys, conditions, forms, or heat treats for each material are identified and described separately in individual records of the file. Each record includes material properties for one to six temperatures covering the normal operating temperature range of that material. An interpolation procedure in the evaluation routine is used to obtain the material properties for temperatures other than those described in the data set.

The material property library was compiled from data developed for various in-house preliminary design and contract proposal efforts. The data are similar to the information found in MIL-HDBK-5A, "Metallic Materials and Elements for Aerospace Vehicle Structures." This document can be used in the preparation of new data sets to expand the scope of the library.

Table 1 lists the materials found in the current SWEEP data bank. To allow for ease in identification, each material is listed by relative record number and descriptive title. This title is always included in the output data set describing the selected structural material for the individual vehicle components being analyzed. This identification of the material used is necessary because material alloy and form along with the source of the data must be easily associated with the structure and weight arrived at in each problem. Material properties at several operating temperatures after specific exposure at temperature are included in this file. These records can be selected when similar requirements are specified for a problem.

TABLE 1. MATERIALS INCLUDED IN THE LIBRARY

| Material ID No. | Material Description | Density ($lb/in^3$) | Basis (Note 1) | Thickness (in.) | Temperature Range (° F) | Room Temp Properties (psi) $F_{CY}$ | Room Temp Properties (psi) $F_{SU}$ |
|---|---|---|---|---|---|---|---|
| 1 | 2024-T81 Al clad sheet | 0.100 | S | 0.063-0.250 | 80 | 57,000 | 39,000 |
| 2 | 2024-T851 Al bare plate | 0.100 | S(2) | 0.500-1.000 | 80-300 | 58,500 | 38,000 |
| 3 | 2024-T851 Al bare plate | 0.100 | S(3) | 1.000-3.000 | 80-350 | 54,500 | 37,500 |
| 4 | 7075-T6 Al clad sheet | 0.101 | B | 0.040-0.062 | 80 | 65,000 | 44,000 |
| 5 | 7075-T6 Al bare plate | 0.101 | B | 0.250-0.500 | 80 | 71,000 | 47,000 |
| 6 | 7075-T6511 Al extrusion | 0.101 | A | 3.000-4.000 | 80 | 66,000 | 45,000 |
| 7 | 7075-T7351 Al bare plate | 0.101 | S | 0.250-0.500 | 80 | 56,000 | 39,000 |
| 8 | 7050-T7351 Al bare plate | 0.102 | Est | - - | 80 | 66,000 | 42,200 |
| 9 | 2219-T851 Al bare sheet/plate | 0.102 | Est | 0.250-2.000 | 80 | 48,000 | 36,000 |
| 10 | 7178-T6 Al clad sheet | 0.102 | B | 0.045-0.249 | 80 | 75,000 | 48,000 |
| 11 | 7178-T6 Al bare sheet | 0.102 | B(4) | 0.045-0.249 | 80-280 | 75,000 | 49,000 |
| 12 | 7079-T651 Al bare plate | 0.099 | A | 0.250-1.500 | 80 | 63,000 | 42,000 |
| 13 | 6Al-4V Ti annealed sheet/plate | 0.160 | B(2) | -0.250 | 80-500 | 138,000 | 81,000 |
| 14 | 6Al-4V Ti annealed plate | 0.160 | S | 0.187-4.000 | 80-350 | 126,000 | 76,000 |
| 15 | 9Ni-4CO-.2C steel sheet/plate | 0.283 | Est | - - | 80 | 188,000 | 118,000 |
| 16 | 17-4PH (H900) | 0.282 | Est | - - | 80 | 165,000 | 120,000 |
| 17 | Rene 41 plate | 0.298 | B | 0.187- | 80-1600 | 113,000 | 118,000 |

NOTE

1. The basis A, B, and S are as defined in MIL-HDBK-5A.
2. After exposure to 290° F for 120 hours.
3. After exposure to 265° F for 390 hours.
4. After exposure to 280° F for 120 hours.

Section II

# METHODOLOGY

## AVAILABLE DATA

Each material properties data set or record consists of the following information:

- Identification number and descriptive title
- Density
- Modulus of elasticity at 80° F
- Fatigue parameter, reduction in area
- Crippling coefficients
- Stress-strain and strength data for different operating temperatures (from one to six temperature blocks can be included)
- Fatigue factors, fraction of ultimate tensile strength

An item-by-item description of these data can be found in Table 2.

## CALCULATION PROCEDURES

### INTERMEDIATE TEMPERATURES

Properties at temperatures other than those input are determined by linear interpolation or extrapolation if the data set has more than one temperature block. For records that include only one temperature, the properties for that temperature are used for all requested temperatures with an accompanying printed message to warn the user.

### STRESS-STRAIN DIAGRAM

Stress-strain data are included at key points, and a continuous description for the desired temperature is provided by a least squares curve fit. Figure 1 shows a typical diagram. The lowest point ($\sigma_{PL}, \epsilon_{PL}$) is the proportional limit, that point where the diagram first deviates from a straight line through the

TABLE 2. DESCRIPTION OF MATERIAL LIBRARY INPUTS BY RELATIVE LOCATION

TITLE CARD ONE 1-80 FOR COMMENTS. **MUST BE USED**
TWO 1-80 FOR COMMENTS. **MUST BE USED**

| LOCATIONS | DESCRIPTIONS | |
|---|---|---|
| 1 | MATERIAL NUMBER-CHECK PROGRAM LIBRARY FOR NEXT NUMBER. | |
| 2 | MATERIAL DENSITY. | POUNDS/CUBIC INCH |
| 3 | YOUNG'S MODULUS AT 80 DEG. FAHRENHEIT | |
| 4 | SHEAR MODULUS AT 80 DEG. FAHRENHEIT | |
| 5 | REDUCTION IN AREA, FATIGUE PARAMETER. | |
| 6 | MATERIAL CRIPPLING COEFFICIENT -FOR FLAT SHEET SIMPLY SUPPORTED AT TWO EDGES | |
| 7 | -FOR FLAT SHEET SIMPLY SUPPORTED AT ONE EDGE ONLY | |
| 8-109 | NOT USED | |
| 110 | TEMPERATURE OF MATERIAL FOR DATA IN LOCATIONS 111 THRU 134 *** TEMPERATURES IN DEGREES FAHRENHEIT *** | |
| 111 | MU - POISSON'S RATIO | |
| 112 | EC1 COMPRESSION STRAIN AT PROPORTIONAL LIMIT | INCH/INCH |
| 113 | EC5 COMPRESSION STRAIN AT YIELD POINT | INCH/INCH |
| 114 | FC1 COMPRESSION STRESS AT PROPORTIONAL LIMIT | PSI |
| 115 | FC2 COMPRESSION STRESS AT EC1 + (EC5 - EC1) * .25 | PSI |
| 116 | FC3 COMPRESSION STRESS AT EC1 + (EC5 - EC1) * .50 | PSI |
| 117 | FC4 COMPRESSION STRESS AT EC1 + (EC5 - EC1) * .75 | PSI |
| 118 | FC5 COMPRESSION STRESS AT YIELD POINT | PSI |
| 119 | ET1 TENSION STRAIN AT PROPORTIONAL LIMIT | INCH/INCH |
| 120 | ET5 TENSION STRAIN AT YIELD POINT | INCH/INCH |
| 121 | FT1 TENSION STRESS AT PROPORTIONAL LIMIT | PSI |
| 122 | FT2 TENSION STRESS AT ET1 + (ET5 - ET1) * .25 | PSI |
| 123 | FT3 TENSION STRESS AT ET1 + (ET5 - ET1) * .50 | PSI |
| 124 | FT4 TENSION STRESS AT ET1 + (ET5 - ET1) * .75 | PSI |
| 125 | FT5 TENSION STRESS AT YIELD POINT | PSI |
| 126 | FTU ULTIMATE TENSION STRESS | PSI |
| 127 | FSU ULTIMATE SHEAR STRESS | PSI |
| 128 | FBRU ULTIMATE BEARING STRESS | PSI |

TABLE 2. DESCRIPTION OF MATERIAL LIBRARY INPUTS BY RELATIVE LOCATION (CONT)

| | |
|---|---|
| 129 | NOT USED |
| 130 | KFTU FATIGUE FACTOR - ENDURANCE LIMIT, KT=1, R=-1 |
| 131 | - FUSELAGE SHELL BENDING |
| 132 | - FUSELAGE PRESSURE CYCLES, R=0 |
| 133 | - WING AT SIDE OF FUSELAGE (SOF) |
| 134 | - WING OUTER PANEL STATION (WOS) |
| | **INPUT FOR SECOND TEMPERATURE** **OPTIONAL** |
| 135 | TEMPERATURE, MUST BE HIGHER THAN TEMPERATURE IN LOCATION 110 |
| 136 | MO. |
| 137 | EC1 STRAIN AT P.L. |
| 138 | EC2 STRAIN AT YIELD |
| 139 | FC1 |
| 140 | FC2 |
| 141 | FC3 |
| 142 | FC4 |
| 143 | FC5 |
| 144 | ET1 STRAIN AT P.L. |
| 145 | ET2 STRAIN AT YIELD |
| 146 | FT1 |
| 147 | FT2 |
| 148 | FT3 |
| 149 | FT4 |
| 150 | FT5 |
| 151 | FTU |
| 152 | FSU |
| 153 | FBRU |
| 154 | NOT USED |
| 155 | KFTU - ENDURANCE LIMIT |
| 156 | - FUS. SHELL |
| 157 | - FUS. PRESSURE |
| 158 | - SOF |
| 159 | - WOS |
| | **INPUT FOR THIRD TEMPERATURE** **OPTIONAL** |
| 160 | TEMPERATURE, MUST BE HIGHER THAN TEMPERATURE IN LOCATION 135 |
| 161 | MO. |

TABLE 5. DESCRIPTION OF MATERIAL LIBRARY INPUTS BY RELATIVE LOCATION (CONT)

| | | |
|---|---|---|
| 162 | EC1 | STRAIN AT P.L. |
| 163 | EC5 | STRAIN AT YIELD |
| 164 | FC1 | |
| 165 | FC2 | |
| 166 | FC3 | |
| 167 | FC4 | |
| 168 | FC5 | |
| 169 | ET1 | STRAIN AT P.L. |
| 170 | ET5 | STRAIN AT YIELD |
| 171 | FT1 | |
| 172 | FT2 | |
| 173 | FT3 | |
| 174 | FT4 | |
| 175 | FT5 | |
| 176 | FTU | |
| 177 | FSU | |
| 178 | FBRU | |
| 179 | NOT USED | |
| 180 | KFTU | - ENDURANCE LIMIT |
| 181 | | - FUS. SHELL |
| 182 | | - FUS. PRESSURE |
| 183 | | - SCF |
| 184 | | - WCS |
| | **INPUT FOR FOURTH TEMPERATURE** **OPTIONAL** | |
| 185 | TEMPERATURE, MUST BE HIGHER THAN TEMPERATURE IN LOCATION 160 | |
| 186 | MU. | |
| 187 | EC1 | STRAIN AT P.L. |
| 188 | EC5 | STRAIN AT YIELD |
| 189 | FC1 | |
| 190 | FC2 | |
| 191 | FC3 | |
| 192 | FC4 | |
| 193 | FC5 | |
| 194 | ET1 | STRAIN AT P.L. |
| 195 | ET5 | STRAIN AT YIELD |
| 196 | FT1 | |

TABLE 2. DESCRIPTION OF MATERIAL LIBRARY INPUTS BY RELATIVE LOCATION (CONT)

| | |
|---|---|
| 197 | FT2 |
| 198 | FT3 |
| 199 | FT4 |
| 200 | FT5 |
| 201 | FTU |
| 202 | FSU |
| 203 | FBRU |
| 204 | NOT USED |
| 205 | KFTU - ENDURANCE LIMIT |
| 206 | - FUS. SHELL |
| 207 | - FUS. PRESSURE |
| 208 | - SCF |
| 209 | - WTS |
| | **INPUT FOR FIFTH TEMPERATURE** **OPTIONAL** |
| 210 | TEMPERATURE, MUST BE HIGHER THAN TEMPERATURE IN LOCATION 185 |
| 211 | MU. |
| 212 | EC1 STRAIN AT P.L. |
| 213 | EC5 STRAIN AT YIELD |
| 214 | FC1 |
| 215 | FC2 |
| 216 | FC3 |
| 217 | FC4 |
| 218 | FC5 |
| 219 | ET1 STRAIN AT P.L. |
| 220 | ET5 STRAIN AT YIELD |
| 221 | FT1 |
| 222 | FT2 |
| 223 | FT3 |
| 224 | FT4 |
| 225 | FT5 |
| 226 | FTU |
| 227 | FSU |
| 228 | FBRU |
| 229 | NOT USED |
| 230 | KFTU - ENDURANCE LIMIT |
| 231 | - FUS. SHELL |

TABLE 2. DESCRIPTION OF MATERIAL LIBRARY INPUTS BY RELATIVE LOCATION (CONCL)

```
232            - FUS. PRESSURE
233            - SUF
234            - WOS

          **INPUT FOR SIXTH TEMPERATURE**     **OPTIONAL**
235       TEMPERATURE, MUST BE HIGHER THAN TEMPERATURE IN LOCATION 210
236       FU.
237       EC1   STRAIN AT P.L.
238       EC5   STRAIN AT YIELD
239       FC1
240       FC2
241       FC3
242       FC4
243       FC5
244       ET1   STRAIN AT P.L.
245       ET5   STRAIN AT YIELD
246       FT1
247       FT2
248       FT3
249       FT4
250       FT5
251       FTU
252       FSU
253       FBRU
254       NOT USED
255       KFTU   - ENDURANCE LIMIT
256              - FUS. SHELL
257              - FUS. PRESSURE
258              - SUF
259              - WOS

*** LAST CARD FOR EACH MATERIAL MUST HAVE A MINUS (-) IN COLUMN ONE(1)

*** LAST MATERIAL MUST BE A DUMMY
      WITH 2 TITLE CARDS AND A 3RD CARD WITH MINUS IN COLUMN 1
                             AND ZERO (0.0) IN LOCATION 1
```

origin. The slope of the straight line portion of the diagram defines the modulus of elasticity, E.

$$E = \sigma_{PL}/\epsilon_{PL} \tag{1}$$

The highest point on the diagram ($\sigma_Y$, $\epsilon_Y$) is the yield point; in this example, it is defined by the 0.002 strain offset method. The true yield stress would be used for materials which have a definite yield point. Three stress values, at equal strain increments between $\epsilon_{PL}$ and $\epsilon_Y$ are included to help define the curved portion of the diagram.

The general form of the equation used to approximate the stress-strain curve is:

$$\epsilon = \frac{\sigma}{E} + ae^{b\sigma} \tag{2}$$

where

$\epsilon$ = strain, inches per inch

$\sigma$ = stress, pounds per square inch

E = modulus of elasticity, pounds per square inch

a = material dependent factor, inches per inch

b = material dependent factor, 1/psi

e = base of natural logarithms

The first term of the equation approximates the linear region of the curve, values below the proportional limit. The second term fits the plastic region of the diagram between the proportional limit and the "yield" point.

For a curve through the proportional limit, yield point, and point B, as shown in Figure 1, the factors a and b can be determined:

$$b = \frac{\ln\left(\frac{\Delta\epsilon_Y}{\Delta\epsilon_B}\right)}{\sigma_Y - \sigma_B} \tag{3}$$

$$a = e^{(\ln \Delta\epsilon_B - b\sigma_B)} \tag{4}$$

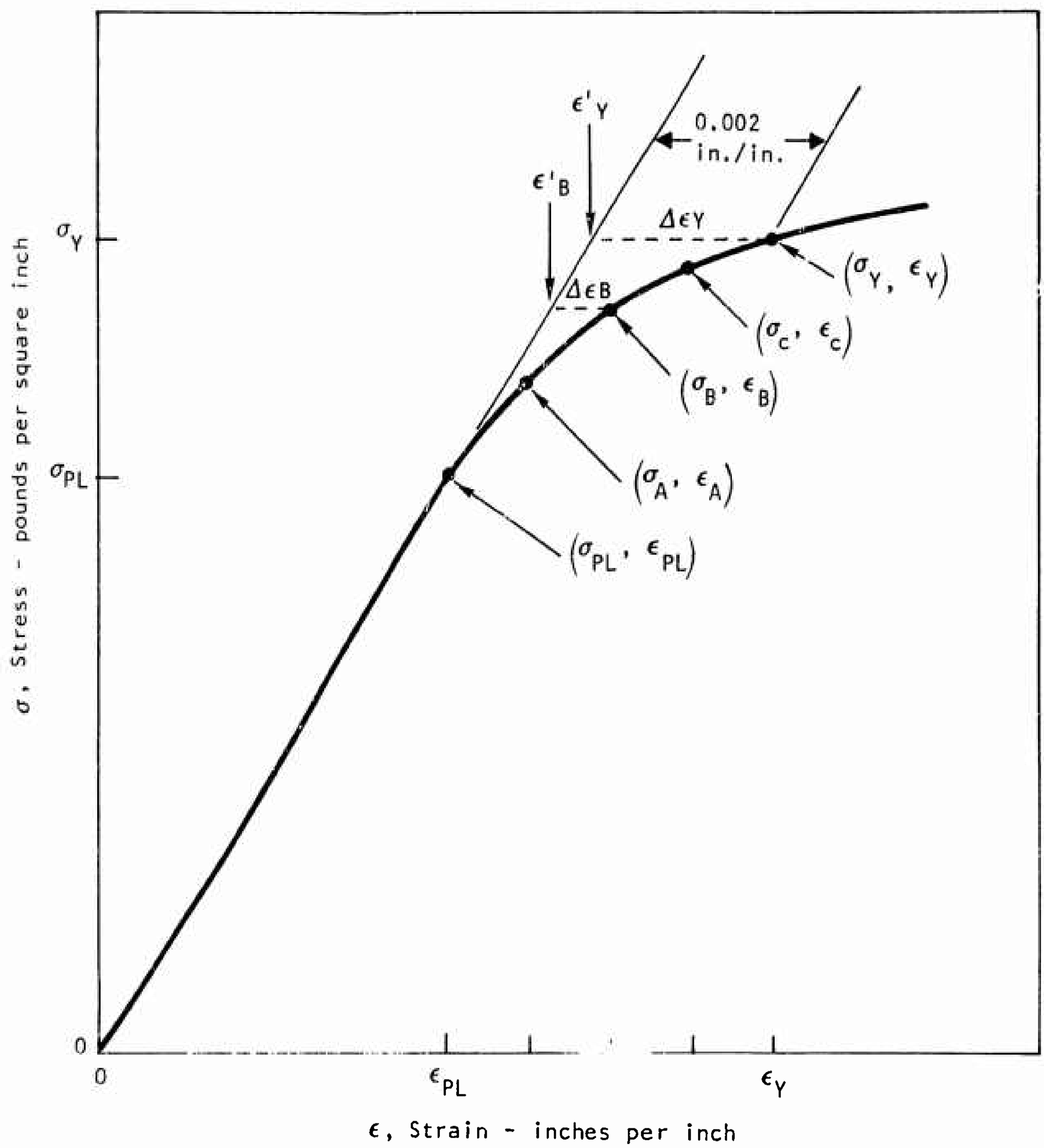



Figure 1. Stress-strain curve showing curve fit control points.

where

$$\Delta\epsilon_y = \epsilon_y - \frac{\sigma_y}{E} = \epsilon_y - \epsilon'_y$$

$$\Delta\epsilon_B = \epsilon_B - \frac{\sigma_B}{E} = \epsilon_B - \epsilon'_B$$

For points A and C, the same procedure is followed, and the best of the three is selected for the calculation of required parameters.

The slope of the curve at any point is the tangent modulus, $E_T$, and is obtained by differentiating equation 2.

$$E_T = \frac{d\sigma}{d\epsilon} = \frac{1}{\frac{d\epsilon}{d\sigma}} = \frac{1}{\frac{1}{E} + abe^{b\sigma}} \tag{5}$$

The secant modulus, $E_S$, is the value of stress divided by strain at any point on the curve.

$$E_S = \sigma/\epsilon \tag{6}$$

These properties are shown on a typical stress-strain diagram in Figure 2.

## OTHER PROPERTIES

Other design properties may be obtained from the library:

- Poisons ratio, $\mu$, used in the calculation of shear modulus:

$$G = E/2(1 + \mu) \tag{7}$$

- Ultimate tensile stress
- Ultimate shear stress
- Ultimate bearing stress

A complete listing of the library data is included in the Users' Manual.

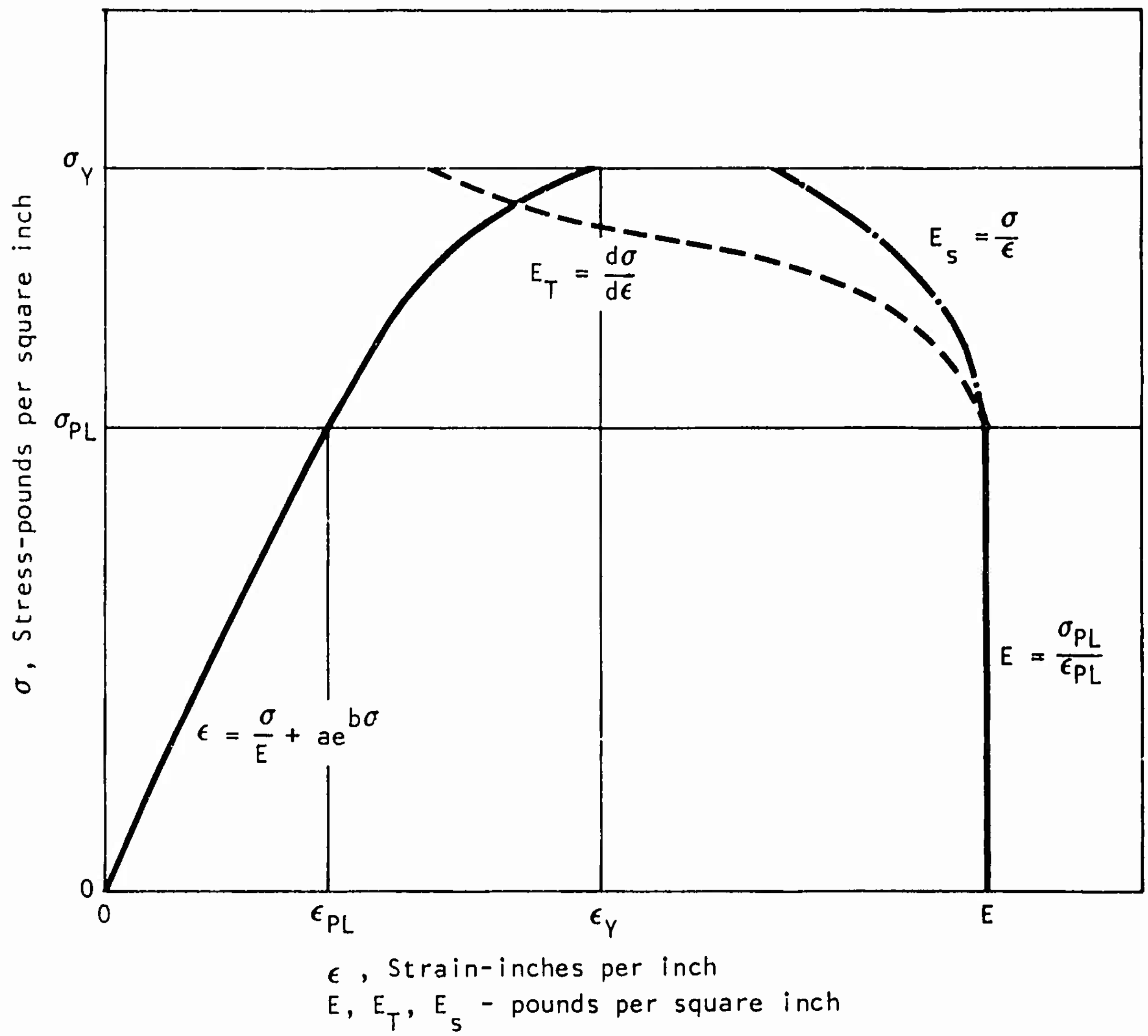



Figure 2. Stress-strain curve evaluation for elastic and plastic properties.

TABLE 5. MATERIAL LIBRARY DATA

MATERIAL NUMBER 1

2024-T81 AL CLAD SHEET 0.063 TO 0.249 IN. MIL-HDBK-5 S DATA EST.
REF. TABLE 3.2.3..0(C) PAGE 258 8-09-72

| | | | | | |
|---|---|---|---|---|---|
| 1 | 1.0 | 0.100 | 10700000.0 | 4022000.0 | .16 |
| 6 | 0.509 | 0.312 | | | |
| 110 | 80.0 | 0.33 | 0.004252336 | 0.007327102 | 45500.0 |
| 115 | 50300.0 | 53600.0 | 55700.0 | 57000.0 | 0.004252336 |
| 120 | 0.007327102 | 45500.0 | 50300.0 | 53600.0 | 55700.0 |
| 125 | 57000.0 | 65000.0 | 39000.0 | 127000.0 | |
| 130 | 0.225 | 0.76 | 0.50 | 1.0 | 1.0 |
| 255 | | | | | |

MATERIAL NUMBER 2

2024-T851 AL BARE PLATE 0.5 TO 1.0 IN. REF-AFL.30/1.10
120 HRS AT 220 DEG MIL-HDBK-5 S DATA 10-24-63

| | | | | | |
|---|---|---|---|---|---|
| 1 | 2.0 | 0.100 | 10700000.0 | 4022560.0 | 0.16 |
| 6 | 0.509 | 0.312 | | | |
| 110 | 80.0 | 0.330 | 0.0043458 | 0.0074673 | 46500.0 |
| 115 | 52100.0 | 55200.0 | 57200.0 | 58500.0 | 0.0043458 |
| 120 | 0.0074673 | 46500.0 | 52100.0 | 55200.0 | 57200.0 |
| 125 | 58500.0 | 66000.0 | 38000.0 | 117000.0 | |
| 130 | 0.225 | 0.76 | 0.50 | 1.0 | 1.0 |
| 135 | 200.0 | 0.345 | 0.004190476 | 0.007285714 | 44000.0 |
| 140 | 49100.0 | 52300.0 | 54200.0 | 55500.0 | 0.004190476 |
| 145 | 0.007285714 | 44000.0 | 49100.0 | 52300.0 | 54200.0 |
| 150 | 55500.0 | 61500.0 | 35500.0 | 109000.0 | |
| 155 | 0.225 | 0.76 | 0.50 | 1.0 | 1.0 |
| 160 | 300.0 | 0.358 | 0.003872549 | 0.006852941 | 39500.0 |
| 165 | 44300.0 | 47000.0 | 48500.0 | 49500.0 | 0.003872549 |
| 170 | 0.006852941 | 39500.0 | 44300.0 | 47000.0 | 48500.0 |
| 175 | 49500.0 | 55000.0 | 31500.0 | 97500.0 | |
| 180 | 0.225 | 0.76 | 0.50 | 1.0 | 1.0 |
| 255 | | | | | |

TABLE 5. MATERIAL LIBRARY DATA (CONT)

MATERIAL NUMBER 3

2024-T851 AL PLATE 1.0 TO 3.0 IN. REF-RA2.8 2-5-70
330 HRS AT 265 DEG. MIL-HDBK-5 S VALUES PER ALCOA 2-09-70

| | | | | | |
|---|---|---|---|---|---|
| 1 | 3.0 | 0.100 | 10770000.0 | 4022560.0 | 1.16 |
| 6 | 0.509 | 0.212 | | | |
| 110 | 80.0 | 0.330 | 0.004159 | 0.007093 | 44500.0 |
| 115 | 49100.0 | 51900.0 | 53500.0 | 54500.0 | 0.004159 |
| 120 | 0.007093 | 44500.0 | 49100.0 | 51900.0 | 53500.0 |
| 125 | 54500.0 | 65000.0 | 37500.0 | 109000.0 | |
| 130 | 0.225 | 0.76 | 0.50 | 1.0 | 1.0 |
| 135 | 165.0 | 0.334 | 0.004095 | 0.007048 | 43000.0 |
| 140 | 47900.0 | 50500.0 | 52000.0 | 53000.0 | 0.004095 |
| 145 | 0.007048 | 43000.0 | 47900.0 | 50500.0 | 52000.0 |
| 150 | 53000.0 | 62500.0 | 36000.0 | 106000.0 | |
| 155 | 0.225 | 0.76 | 0.50 | 1.0 | 1.0 |
| 160 | 197.0 | 0.336 | 0.004048 | 0.006952 | 42500.0 |
| 165 | 47100.0 | 49700.0 | 51200.0 | 52000.0 | 0.004048 |
| 170 | 0.006952 | 42500.0 | 47100.0 | 49700.0 | 51200.0 |
| 175 | 52000.0 | 61000.0 | 35000.0 | 104000.0 | |
| 180 | 0.225 | 0.76 | 0.50 | 1.0 | 1.0 |
| 185 | 265.0 | 0.339 | 0.003883 | 0.006757 | 40000.0 |
| 190 | 44500.0 | 46900.0 | 48200.0 | 49000.0 | 0.003883 |
| 195 | 0.006757 | 40000.0 | 44500.0 | 46900.0 | 48200.0 |
| 200 | 49000.0 | 56500.0 | 32500.0 | 98000.0 | |
| 205 | 0.225 | 0.76 | 0.50 | 1.0 | 1.0 |
| 210 | 350.0 | 0.343 | 0.003550 | 0.006300 | 35500.0 |
| 215 | 39100.0 | 41500.0 | 42500.0 | 43000.0 | 0.003550 |
| 220 | 0.006300 | 35500.0 | 39100.0 | 41500.0 | 42500.0 |
| 225 | 43000.0 | 49000.0 | 28000.0 | 86000.0 | |
| 230 | 0.225 | 0.76 | 0.50 | 1.0 | 1.0 |
| 255 | | | | | |

TABLE 5. MATERIAL LIBRARY DATA (CONT'D)

```
MATERIAL NUMBER  4

     7075-T6 AL CLAD SHEET 0.040 TO 0.062 IN. MIL-HDBK-5 B DATA EST.
                 REF. TABLE 3.2.7.0(C) PAGE 336  8-19-72
       1 4.0          0.101         10700000.0  4022560.0    0.18
       6 0.509        0.312
     110 80.0         0.3305        0.00380952  0.00819048   40000.0
     115 51200.0      59000.0       52000.0     65000.0      0.00380952
     120 0.00819048   40000.0       51200.0     59000.0      62000.0
     125 65000.0      73000.0       44000.0     133000.0
     130 0.225        0.76          0.500       1.0          1.0
     255
```

```
MATERIAL NUMBER  5

     7075-T6 AL BARE PLATE 0.25 TO 0.50 IN. MIL-HDBK-5 B DATA EST.
                 REF. TABLE 3.2.7.0(B) PAGE 334  4-06-72
       1 5.0          .101          10500000.0  3900000.0    0.18
       6 0.509        0.312
     110 80.0         .33           .005        .0087619     52500.0
     115 59700.0      65000.0       68750.0     71000.0      .005
     120 .0087619     52500.0       59700.0     65000.0      68750.0
     125 71000.0      79000.0       47000.0     142000.0
     130 0.225        0.76          0.500       1.0          1.0
     255
```

TABLE 5. MATERIAL LIBRARY DATA (CONT)

MATERIAL NUMBER 6

7075-T6511 AL EXTRU. 3.0 TO 4.0 IN. MIL-HDBK-5 A DATA EST.
REF. TABLE 3.2.7.0(F) PAGE 340 2-26-72

| | | | | | |
|---|---|---|---|---|---|
| 1 | 6.0 | 0.101 | 10500000.0 | 3900000.0 | 0.18 |
| 6 | 0.509 | 0.312 | | | |
| 110 | 80.0 | 0.330 | 0.005142857 | 0.008285714 | 54000.0 |
| 115 | 59100.0 | 62300.0 | 64450.0 | 66000.0 | 0.005142857 |
| 120 | 0.008285714 | 54000.0 | 59100.0 | 62300.0 | 64450.0 |
| 125 | 66000.0 | 81000.0 | 45000.0 | 97000.0 | |
| 130 | 0.225 | 0.76 | 0.500 | 1.0 | 1.0 |
| 255 | | | | | |

---

MATERIAL NUMBER 7

7075-T7351 AL BARE PLATE 0.25 TO 0.50 IN. MIL-HDBK-5 S DATA EST.
REF. TABLE 3.2.7.0(B) PAGE 334 12-14-71

| | | | | | |
|---|---|---|---|---|---|
| 1 | 7.0 | .101 | 10500000.0 | 3900000.0 | 0.18 |
| 6 | 0.509 | 0.312 | | | |
| 110 | 80.0 | .33 | .0043204 | .0074369 | 44500.0 |
| 115 | 49800.0 | 52800.0 | 54700.0 | 56000.0 | .0043204 |
| 120 | .0074369 | 44500.0 | 49800.0 | 52800.0 | 54700.0 |
| 125 | 56000.0 | 63000.0 | 39000.0 | 137000.0 | |
| 130 | 0.225 | 0.76 | 0.500 | 1.0 | 1.0 |
| 255 | | | | | |

TABLE 5. MATERIAL LIBRARY DATA (CONT)

MATERIAL NUMBER 8

7050-T7351 AL BARE PLATE
ESTIMATED DESIGN VALUE 2 MAY 1972

| | | | | | |
|---|---|---|---|---|---|
| 1 | 9.0 | 0.102 | 10500000.0 | 3900000.0 | 0.20 |
| 6 | 0.500 | 0.312 | | | |
| 110 | 80.0 | 0.33 | 0.005029 | 0.008301 | 52800.0 |
| 115 | 59200.0 | 67000.0 | 55100.0 | 66000.0 | 0.005029 |
| 120 | 0.008301 | 52800.0 | 58200.0 | 63000.0 | 65100.0 |
| 125 | 66000.0 | 76000.0 | 42200.0 | 97000.0 | |
| 130 | 0.225 | 0.76 | 0.500 | 1.0 | 1.0 |
| 255 | | | | | |

MATERIAL NUMBER 9

2219-T851 AL BARE SHEET AND PLATE 0.25 TO 2.0 IN.
CURVE DERIVED FROM T852 CURVE AND T851 CHART 12 JUNE 1972

| | | | | | |
|---|---|---|---|---|---|
| 1 | 9.0 | .102 | 10500000.0 | 4000000.0 | 0.27 |
| 6 | 0.500 | 0.312 | | | |
| 110 | 80.0 | .33 | .00354 | .00646 | 33360.0 |
| 115 | 43800.0 | 46300.0 | 47500.0 | 48000.0 | .00364 |
| 120 | .00646 | 39360.0 | 43800.0 | 46300.0 | 47500.0 |
| 125 | 48000.0 | 62000.0 | 36000.0 | 118000.0 | |
| 130 | 0.225 | 0.76 | 0.500 | 1.0 | 1.0 |
| 255 | | | | | |

TABLE 5. MATERIAL LIBRARY DATA (CONT)

MATERIAL NUMBER 10

7178-T6 AL CLAD SHEET 0.045 TO 0.249 IN. MIL-HDBK-5 B DATA EST.
REF. TABLE 3.7.9.0(C) PAGE 369 9-09-72

| | | | | | |
|---|---|---|---|---|---|
| 1 | 10.0 | 0.102 | 10500000.0 | 3900000.0 | 0.20 |
| 6 | 0.509 | 0.312 | | | |
| 110 | 80.0 | 0.33 | 0.004904761 | 0.009142857 | 51500.0 |
| 115 | 61000.0 | 67800.0 | 72500.0 | 75000.0 | 0.004904761 |
| 120 | 0.009142857 | 51500.0 | 61000.0 | 67800.0 | 72500.0 |
| 125 | 75000.0 | 80000.0 | 48000.0 | 152000.0 | |
| 130 | 0.225 | 0.76 | 0.50 | 1.0 | 1.0 |
| 255 | | | | | |

---

MATERIAL NUMBER 11

7178-T6 AL BARE SHEET 0.045 TO 0.249 IN. REF-FX-45 1-25-68
120 HRS AT 280 DEG. MIL-HDBK-5 B DATA

| | | | | | |
|---|---|---|---|---|---|
| 1 | 11.0 | 0.102 | 10500000.0 | 3900000.0 | 0.20 |
| 6 | 0.509 | 0.312 | | | |
| 110 | 80.0 | 0.330 | 0.004904761 | 0.009142857 | 51500.0 |
| 115 | 61000.0 | 67800.0 | 72500.0 | 75000.0 | 0.004904761 |
| 120 | 0.009142857 | 51500.0 | 61000.0 | 67800.0 | 72500.0 |
| 125 | 75000.0 | 81000.0 | 49000.0 | 153500.0 | |
| 130 | 0.225 | 0.76 | 0.500 | 1.0 | 1.0 |
| 135 | 280.0 | 0.355 | 0.005157894 | 0.008736842 | 49000.0 |
| 140 | 54600.0 | 58400.0 | 61400.0 | 64000.0 | 0.005157894 |
| 145 | 0.008736842 | 49000.0 | 54600.0 | 58400.0 | 61400.0 |
| 150 | 64000.0 | 59500.0 | 46000.0 | 113000.0 | |
| 155 | 0.225 | 0.76 | 0.50 | 1.0 | 1.0 |
| 255 | | | | | |

TABLE 5. MATERIAL LIBRARY DATA (CONT)

MATERIAL NUMBER 12

7079-T651 AL BARE PLATE 0.25 TO 1.50 IN. MIL-HDBK-5 A DATA EST.
REF. TABLE 3.2.8.0(D) PAGE 358 2-24-72

| | | | | | |
|---|---|---|---|---|---|
| 1 | 12.0 | 0.098 | 10500000.0 | 3900000.0 | 0.20 |
| 6 | 0.509 | 0.312 | | | |
| 110 | 80.0 | 0.330 | 0.0043810 | 0.008000 | 52800.0 |
| 115 | 52800.0 | 57900.0 | 61100.0 | 63000.0 | 0.0043810 |
| 120 | 0.0080 | 46000.0 | 52800.0 | 57900.0 | 61100.0 |
| 125 | 63000.0 | 71000.0 | 42000.0 | 114000.0 | |
| 130 | 0.225 | 0.76 | 0.500 | 1.0 | 1.0 |
| 255 | | | | | |

TABLE 5. MATERIAL LIBRARY DATA (CONT)

MATERIAL NUMBER 13

6AL-4V TI-A* SHT/PLATE TO .250 IN. REF-TF1.30/1.10 2-22-63
120 HRS AT 290 DEG. MIL-HDBK-5 B DATA

| | | | | | |
|---|---|---|---|---|---|
| 1 | 13.0 | 0.160 | 16400220.0 | 6165500.0 | 0.27 |
| 6 | 0.771 | 0.304 | | | |
| 110 | 80.0 | 0.330 | 0.007256 | 0.0104146 | 113000.0 |
| 115 | 127000.0 | 132000.0 | 135500.0 | 138000.0 | 0.007256 |
| 120 | 0.0104146 | 119000.0 | 127000.0 | 132000.0 | 135500.0 |
| 125 | 138000.0 | 132000.0 | 81000.0 | 250000.0 | |
| 130 | 0.28 | 0.70 | 0.50 | 1.0 | 1.0 |
| 135 | 200.0 | 0.336 | 0.0065397 | 0.0099365 | 103000.0 |
| 140 | 111600.0 | 113500.0 | 122200.0 | 125000.0 | 0.0065397 |
| 145 | 0.0099365 | 103000.0 | 111600.0 | 118500.0 | 122200.0 |
| 150 | 125000.0 | 126000.0 | 74500.0 | 250000.0 | |
| 155 | 0.28 | 0.70 | 0.50 | 1.0 | 1.0 |
| 160 | 245.0 | 0.33825 | 0.0065326 | 0.0097813 | 100975.0 |
| 165 | 109000.0 | 114400.0 | 117900.0 | 120275.0 | 0.0065326 |
| 170 | 0.0097813 | 100975.0 | 109000.0 | 114400.0 | 117900.0 |
| 175 | 120275.0 | 122500.0 | 72250.0 | 240550.0 | |
| 180 | 0.28 | 0.70 | 0.50 | 1.0 | 1.0 |
| 185 | 300.0 | 0.3410 | 0.0065232 | 0.0095828 | 98500.0 |
| 190 | 105300.0 | 109600.0 | 112600.0 | 114500.0 | 0.0065232 |
| 195 | 0.0095828 | 98500.0 | 105300.0 | 109600.0 | 112600.0 |
| 200 | 114500.0 | 117000.0 | 69500.0 | 229000.0 | |
| 205 | 0.28 | 0.70 | 0.50 | 1.0 | 1.0 |
| 210 | 500.0 | 0.351 | 0.0060932 | 0.0090609 | 85000.0 |
| 215 | 91200.0 | 95100.0 | 97400.0 | 98500.0 | 0.0060932 |
| 220 | 0.0090609 | 85000.0 | 91200.0 | 95100.0 | 97400.0 |
| 225 | 98500.0 | 107000.0 | 62500.0 | 209427.0 | |
| 230 | 0.28 | 0.70 | 0.50 | 1.0 | 1.0 |
| 255 | | | | | |

TABLE 5. MATERIAL LIBRARY DATA (CONT)

MATERIAL NUMBER 14

6AL-4V TI-A*PLATE 3/16 TO 4.0 IN. REF- SDM IIB-9.5.1.3.11
9-13-72 390 HRS AT 265 DEG. MIL-HDBK-5 S DATA

| | | | | | |
|---|---|---|---|---|---|
| 1 | 14.0 | 0.16 | 16400000.0 | 6165790.0 | 0.27 |
| 6 | 0.771 | 0.304 | | | |
| 10 | 0.040 | 0.332 | 0.030 | 0.040 | 0.080 |
| 110 | 80.0 | 0.33 | 0.0064634 | 0.0096829 | 106000.0 |
| 115 | 115000.0 | 121000.0 | 124000.0 | 126000.0 | 0.0064634 |
| 120 | 0.0096829 | 106000.0 | 115000.0 | 121000.0 | 124000.0 |
| 125 | 126000.0 | 130000.0 | 76000.0 | 245000.0 | |
| 130 | 0.23 | 0.70 | 0.50 | 1.0 | 1.0 |
| 135 | 165.0 | 0.334 | 0.0061635 | 0.00935849 | 98000.0 |
| 140 | 104000.0 | 112100.0 | 115300.0 | 117000.0 | 0.0061635 |
| 145 | 0.00935849 | 98000.0 | 104000.0 | 112100.0 | 115300.0 |
| 150 | 117000.0 | 121000.0 | 71500.0 | 234000.0 | |
| 155 | 0.23 | 0.70 | 0.50 | 1.0 | 1.0 |
| 160 | 197.0 | 0.336 | 0.00608280 | 0.00926114 | 95500.0 |
| 165 | 103700.0 | 108900.0 | 112200.0 | 114000.0 | 0.00608280 |
| 170 | 0.00926114 | 95500.0 | 103700.0 | 108900.0 | 112200.0 |
| 175 | 114000.0 | 118000.0 | 70000.0 | 228000.0 | |
| 180 | 0.23 | 0.70 | 0.50 | 1.0 | 1.0 |
| 185 | 265.0 | 0.339 | 0.005882 | 0.0089934 | 90000.0 |
| 190 | 98300.0 | 102500.0 | 105000.0 | 107000.0 | 0.005882 |
| 195 | 0.0089934 | 90000.0 | 98300.0 | 102500.0 | 105000.0 |
| 200 | 107000.0 | 112500.0 | 67000.0 | 216000.0 | |
| 205 | 0.23 | 0.70 | 0.50 | 1.0 | 1.0 |
| 210 | 350.0 | 0.343 | 0.0057432 | 0.0088243 | 85000.0 |
| 215 | 93500.0 | 97900.0 | 100000.0 | 101000.0 | 0.0057432 |
| 220 | 0.0088243 | 85000.0 | 93500.0 | 97900.0 | 100000.0 |
| 225 | 101000.0 | 106500.0 | 64000.0 | 202000.0 | |
| 230 | 0.23 | 0.70 | 0.50 | 1.0 | 1.0 |
| 255 | | | | | |

TABLE 5. MATERIAL LIBRARY DATA (CONT)

MATERIAL NUMBER 15

MATL NO 15     9NI-4CO-.2C STEEL   REF. TTB-9 MATERIALS MANUAL
BAR,SHEET,PLATE,FORGING

| | | | | | |
|---|---|---|---|---|---|
| 1 | 15.0 | 0.283 | 29600000.0 | 11100000.0 | 0.65 |
| 6 | 0.750 | 0.300 | | | |
| 110 | 80.0 | 0.30 | 0.004700 | 0.008351 | 139170.0 |
| 115 | 160500.0 | 174500.0 | 183000.0 | 198000.0 | 0.004700 |
| 120 | 0.008351 | 139170.0 | 160500.0 | 174500.0 | 183000.0 |
| 125 | 198000.0 | 190000.0 | 118000.0 | 298000.0 | |
| 130 | 0.28 | 0.70 | 0.50 | 1.0 | 1.0 |
| 255 | | | | | |

MATERIAL NUMBER 16

MATL NO 16   17-4 PH STEEL   BAR,FORGING   REF.BRJHN TABLE 81

| | | | | | |
|---|---|---|---|---|---|
| 1 | 16.0 | 0.282 | 27500000.0 | 11000000.0 | 0.45 |
| 6 | 0.631 | 0.333 | | | |
| 110 | 80.0 | 0.272 | 0.004727 | 0.008000 | 130000.0 |
| 115 | 148000.0 | 157000.0 | 162000.0 | 165000.0 | 0.004727 |
| 120 | 0.008000 | 130000.0 | 148000.0 | 157000.0 | 162000.0 |
| 125 | 165000.0 | 180000.0 | 120000.0 | 300000.0 | |
| 130 | 0.28 | 0.70 | 0.50 | 1.0 | 1.0 |
| 255 | | | | | |

TABLE 5. MATERIAL LIBRARY DATA (CONT)

MATERIAL NUMBER 17

MATL NO 17 RENE 41 PLATE T. IS GREATER THAN 0.187 4/2/73
DATA ESTIMATED FROM MIL-HDBK-5B PP. 6-76 TO 6-81 EXPOSURE UP TO 1

| | | | | | |
|---|---|---|---|---|---|
| 1 | 17.0 | 0.298 | 31600000.0 | 12100000.0 | 0.65 |
| 6 | 0.700 | 0.300 | | | |
| 110 | 80.0 | 0.31 | 0.00275 | 0.0056 | 86000.0 |
| 115 | 97000.0 | 103800.0 | 109000.0 | 113000.0 | 0.00275 |
| 120 | 0.0056 | 86000.0 | 97000.0 | 103800.0 | 109000.0 |
| 125 | 113000.0 | 170000.0 | 118000.0 | 253000.0 | |
| 130 | 0.28 | 0.7 | 0.5 | 1.0 | 1.0 |
| 135 | 400.0 | 0.31 | 0.0028 | 0.0057 | 83000.0 |
| 140 | 94000.0 | 101000.0 | 106000.0 | 109500.0 | 0.0028 |
| 145 | 0.0057 | 83000.0 | 94000.0 | 101000.0 | 106000.0 |
| 150 | 109500.0 | 158000.0 | 107400.0 | 240000.0 | |
| 155 | 0.28 | 0.7 | 0.5 | 1.0 | 1.0 |
| 160 | 800.0 | 0.31 | 0.003 | 0.0059 | 82000.0 |
| 165 | 93000.0 | 100000.0 | 104500.0 | 108000.0 | 0.003 |
| 170 | 0.0059 | 82000.0 | 93000.0 | 100000.0 | 104500.0 |
| 175 | 108000.0 | 153000.0 | 106000.0 | 227700.0 | |
| 180 | 0.28 | 0.7 | 0.5 | 1.0 | 1.0 |
| 185 | 1200.0 | 0.31 | 0.0033 | 0.0063 | 81000.0 |
| 190 | 92000.0 | 98500.0 | 103000.0 | 107000.0 | 0.0033 |
| 195 | 0.0063 | 81000.0 | 92000.0 | 98500.0 | 103000.0 |
| 200 | 107000.0 | 148000.0 | 105200.0 | 215000.0 | |
| 205 | 0.28 | 0.7 | 0.5 | 1.0 | 1.0 |
| 210 | 1400.0 | 0.31 | 0.003 | 0.0058 | 67000.0 |
| 215 | 75500.0 | 81000.0 | 85000.0 | 88000.0 | 0.003 |
| 220 | 0.0058 | 67000.0 | 75500.0 | 81000.0 | 85000.0 |
| 225 | 88000.0 | 127500.0 | 105000.0 | 202000.0 | |
| 230 | 0.28 | 0.7 | 0.5 | 1.0 | 1.0 |
| 235 | 1600.0 | 0.31 | 0.0022 | 0.00475 | 37000.0 |
| 240 | 43000.0 | 46000.0 | 47800.0 | 48500.0 | 0.0022 |
| 245 | 0.00475 | 37000.0 | 43000.0 | 46000.0 | 47800.0 |
| 250 | 48500.0 | 86700.0 | 64900.0 | 121400.0 | |
| 255 | 0.28 | 0.7 | 0.5 | 1.0 | 1.0 |

TABLE 3. MATERIAL LIBRARY DATA (CONCL)

```
MATERIAL NUMBER 00 --- DUMMY --- TO END LOAD

     DUMMY      DUMMY
 LAST MATL
-          1 0.0
```

## Section III

## USING THE LIBRARY

### GENERAL DISCUSSION

The synthesis modules use material properties in many equations in the determination of sizing and weight. To obtain the needed factors, each module contains some routines that access the library and perform the required calculations. It is these routines which will be reviewed in this section; a summary list is given in Table 4.

### INPUT PROCEDURE

The material properties input decks are part of the permanent data. These data form the second file on the program tape. During the loading process, the permanent data are transferred to a file called TAPE7, from which they are read in ordered sequence into various records in storage by program READ. These records can be accessed by any routine and can be reinitialized by READ for succeeding cases if the user so requests.

The material properties are records 41 through 60 in storage.

TABLE 4. ROUTINES USING MATERIAL LIBRARY

| Routine | Module | Function Related to Library |
|---|---|---|
| READ | Input data processing | Initialize and reinitialize all material records. |
| WHVMAT | Flutter-temperature | Read records for wing, horizontal and vertical. |
| FTGCTL | Fatigue | Read records for wing, fuselage cover, and fuselage minor frames.<br>Calculate $KF_{TU}$'s for those read.<br>Rewrite records read. |
| MCNTL1 | Air induction systems | Read records for ducts, ramps, and nacelles. |
| MATLF1 | Air induction systems | Calculate required parameters. |
| MATLP2 | Air induction systems | Print. |
| MTLCW | Wing and empennage | Read for wing (and pivot) or horizontal or vertical, whichever in work. |
| MTLFW | Wing and empennage | Calculate required parameters. |
| MTLPW | Wing and empennage | Print. |
| MFCNTL | Fuselage | Read for cover, longeron, major frames, and minor frames. |
| MATLF | Fuselage | Calculate. |
| MATLP1 | Fuselage | Print. |

## FLUTTER AND TEMPERATURE MODULE

In this module, the routine WINMAT reads the material records for the wing, horizontal tail, and vertical tail and composes a table of ultimate tensile stress and shear modulus versus temperature. From each material record, the parameters used are as follows:

### MATERIAL PROPERTIES USED

| Description | Relative Location |
|---|---|
| Temperature | 110, 135, 160, ...235 |
| $\mu$, Poisson's Ratio | 111, 136, 161, ...236 |
| $\epsilon_{CPL}$, compression strain at proportional limits | 112, 137, 162, ...237 |
| $\sigma_{CPL}$, compression stress at proportional limit | 114, 139, 164, ...239 |
| $F_{cy}$, compression yield stress | 118, 143, 168, ...243 |

## FATIGUE MODULE

The fatigue control subroutine, FTGCTL, reads and writes those material records for which the calculation of fatigue factor, $KF_{TU}$, has been requested. This is the only place where material library data is changed from its input values. Fatigue calculations can be performed for two stations on the wing and for the fuselage cover and minor frames.

### MATERIAL PROPERTIES USED

| Description | Relative location |
|---|---|
| RA, reduction in area, fatigue parameter | 5 |

$$RA = \frac{A_i - A_f}{A_i}$$

where

$A_i$ = initial specimen cross-section area

$A_f$ = final fracture cross-section area

| | |
|---|---|
| $\epsilon_{CPL}$, compression strain at proportional limit | 112 |
| $\sigma_{CPL}$, compression stress at proportional limit | 114 |
| $F_{tu}$, ultimate tension stress | 126 |

## MATERIAL PROPERTIES CALCULATED

| Description | Relative location |
|---|---|
| $KF_{TU}$, for fuselage endurance limit | 130, 155, 180 ...255 |
| $KF_{TU}$, for fuselage pressure cycles | 132, 157, 182 ...257 |
| $KF_{TU}$, for wing at side of fuselage | 133, 158, 183 ...258 |
| $KF_{TU}$, for wing at outboard station | 134, 159, 184 ...259 |

If any of the possible calculations are not requested, or if the calculation fails, the location in the library for that one is left unchanged. The material indentification numbers and the before-and-after values of $KF_{TU}$ are printed at the end of the calculation.

## AIR INDUCTION SYSTEM MODULE

This module includes three routines which handle material library data. MCNTLI is the control routine which reads library records for duct, ramp, and nacelle materials, as applicable for the vehicle in work. In MATLF1, the desired properties are obtained for temperatures corresponding to select points on the speed profile. The third routine MATLP2 is called only if a printout of the interpolated material properties is requested.

## WING AND EMPENNAGE MODULE

The material control routine for this module is MTLCW. It reads the library record for the wing (or horizontal or vertical) material and calls MTLFW. There the interpolation to design temperature and the stress-strain curve fits are performed. On return to MTLCW, the required material properties are saved in common for use by the synthesis routines. The subroutine MTLPW is called only if a printout of the developed properties is requested. If the wing data specifies a pivot, the procedure is repeated for the pivot material.

## FUSELAGE MODULE

In control routine MFCNTL, library records are read sequentially for four material numbers designating major frame, minor frame, longeron, and cover materials. For each one, the material properties, interpolated to the temperature for a specific load condition, are determined in MATLF. The interpolated values for the four are saved in a record by MFCNTL for later use by the structural routines. The routine MATLP1 is called if a printout of the calculated material properties is requested.

# PART 2

# FLUTTER AND TEMPERATURE MODULE

## Section I

## INTRODUCTION AND SUMMARY

### PROGRAM OBJECTIVES

The objectives of the flutter and temperature module are to set up the following data:

1. Structural temperatures corresponding to loading conditions. These are used by the synthesis modules in sizing wing, tail, body, and air induction system structures.

2. Data used by the wing and empennage modules in performing flutter analyses. These include the following:

   a. Definition of the critical lifting surface flutter design points (Mach number and altitude), and the corresponding structural temperatures and shear moduli

   b. Definition of the dynamic pressures which are used in the manner of equivalent incompressible subsonic dynamic pressure for torsional divergence. These are for use in the wing and empennage module flutter routines.

The flutter and temperature module does not size or synthesize any of the structure. It does not perform the flutter analysis, which is a function of the wing and empennage modules.

### MODULE STRUCTURE AND OPERATION

The program is written in FORTRAN IV extended program language for operation on the CDC 6600 computer, and is structured as a single overlay within 50,000 octal core locations.

The module is executed as part of the integrated structural weight estimation program (SWEEP). The module is automatically executed as part of the data management and design criteria generation sequence.

### MODULE INPUT AND OUTPUT

Input data are initially set up by the data management module and transferred to the flutter and temperature module via mass storage and labeled common.

Output, consisting of speed-altitude profile data, structural temperatures, materials properties data, dynamic pressures, and inertia data, are transferred to the airloads and the wing and empennage modules via mass storage and labeled common.

## APPROACH TO TEMPERATURE EVALUATION

### THE NATURE OF THERMAL PREDICTIONS IN PRELIMINARY DESIGN

The prediction of temperature for a new design of aircraft is an iterative process. The general characteristics and performance levels permit first approximations from which designers can make initial selections of materials, arrangements, and structural concepts. These selections give the thermal analyst enough information to furnish the designer and structural analyst with improved and extended predictions. This interchange of information is repeated as the design progresses. It is applied to successively smaller details as detailed information becomes available, until finally an airplane evolves that is compatible with its natural and induced environments.

As a tool for preliminary design, this program will be used early in this iterative process when highly detailed estimates of temperature are not needed, and when available information is insufficient for such estimates. A sophisticated thermal analyzer program would be undesirable, as it would add unneeded complexity and increased core storage. The thermal routine is of sufficient depth for this phase of design.

### THE USE OF SKIN TEMPERATURES

Normally, most aircraft structure is in close proximity to the skin both physically and thermally. The thermal resistance across structural joints is small; for instance, the temperatures of a skin and its supporting frame are close. Metals have high thermal conductivity, and the various frames and stiffeners provide good paths for heat flow to the interior. Heat to or from structural elements must enter or leave the airplane through the skin. Thus, the temperature of the substructure follows the temperature of the skin and approaches it under steady-state flight. The usual practice in preliminary design is to use skin temperature for structural temperature, and this practice is followed in this program. However, this generality must be treated with discretion, as there are many exceptions. What is acceptable for preliminary sizing, material selection, and weight estimates is not acceptable for detailed design.

## LIMITATIONS AND ACCURACY

This thermal routine is a so-called "steady-state program;" that is, it gives the temperature which would exist after equilibrium has been established or, conversely, it gives the temperatures which would exist instantaneously in transient flight if the structure had zero heat capacity. Most structural skins approach their equilibrium temperatures rapidly, as their heat capacities are small compared to aerodynamic heating. Exceptions are skins over sources or sinks of heat such as hot turbojet engines or cold integral fuel tanks. The use of the steady-state format eliminates an incompatibility of structural and thermal analyses; the structural analysis is concerned with what is happening to the airplane at a particular instant; the thermal analysis is concerned with what has happened during the immediately prior history.

The shortcoming of a steady-state analysis is that actual temperatures lag behind those computed with zero heat capacity. This shortcoming may be appreciable when the critical airloads are applied immediately after transient maneuvers. It is of little concern when critical airloads are applied after several minutes of flight at the same altitude and mach number. The latter applies to much of preliminary design.

Figure 3 compares temperatures from a transient analysis (considering the heat capacity) with those from a so-called steady-state analysis where the heat capacity is ignored. Here, a heavy skin (0.190-inch thick) has been cold soaked during prolonged subsonic flight at medium altitude. At time zero, the airplane starts a rapid acceleration and climb to supersonic speed at high altitude. It maintains the speed and altitude for three-tenths of an hour. Then it descends and decelerates to medium altitude and loiter speed. Notice that:

1. The maximum temperatures are almost the same.

2. The general trends are the same

3. Error can result at any one instant of time by using the dashed curve.

4. For lighter skin thickness, the two curves would be closer to coincidence.

The limitations of this method lie in applications as previously discussed. Where structure is in close proximity to the skin and when the flight conditions are not transient, it is quite accurate. This is illustrated by Figure 4 which compares computed and measured temperatures on the XB-70 after 32 minutes of flight at mach 3.0.

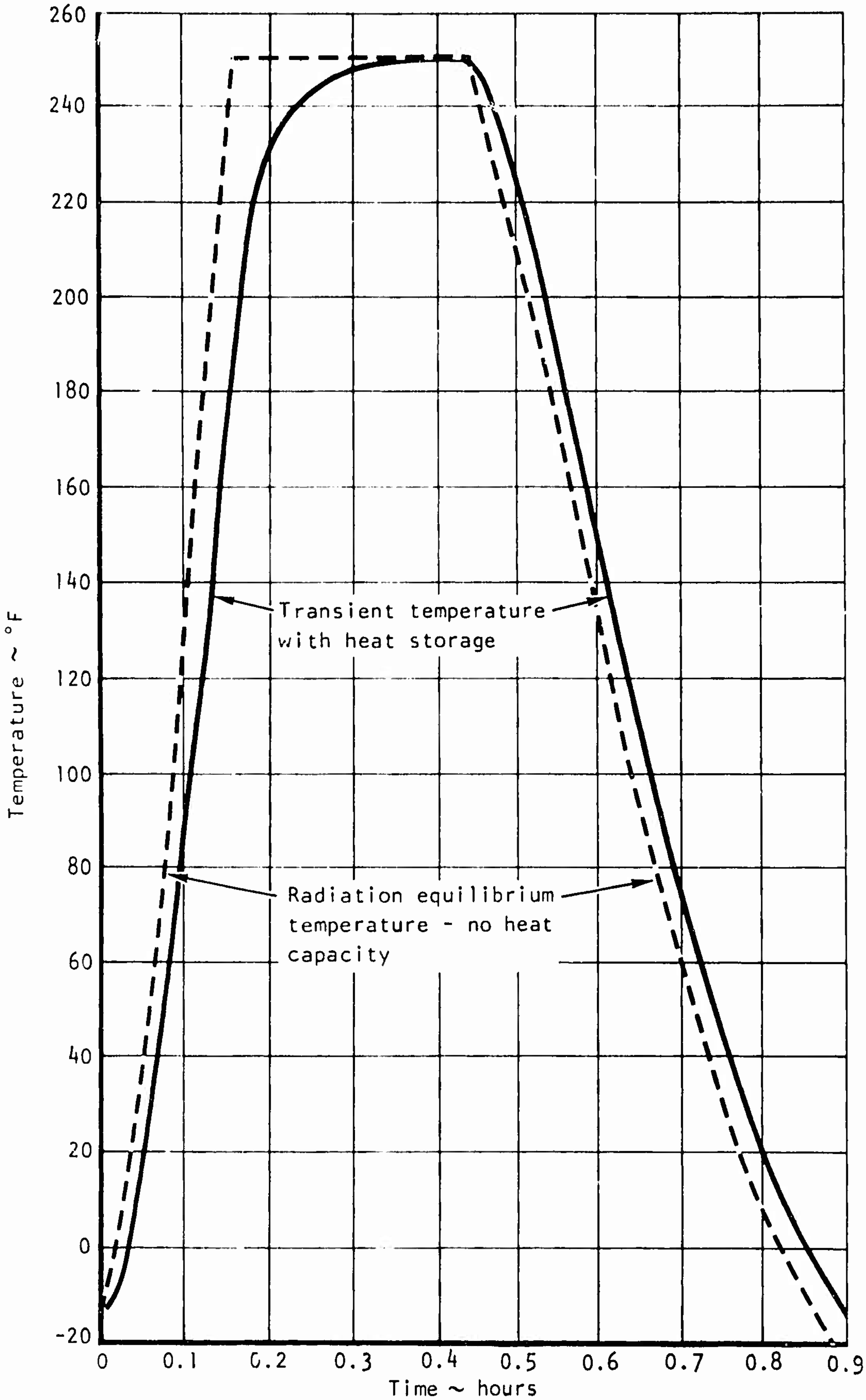



Figure 3. Effect of ignoring heat capacity.

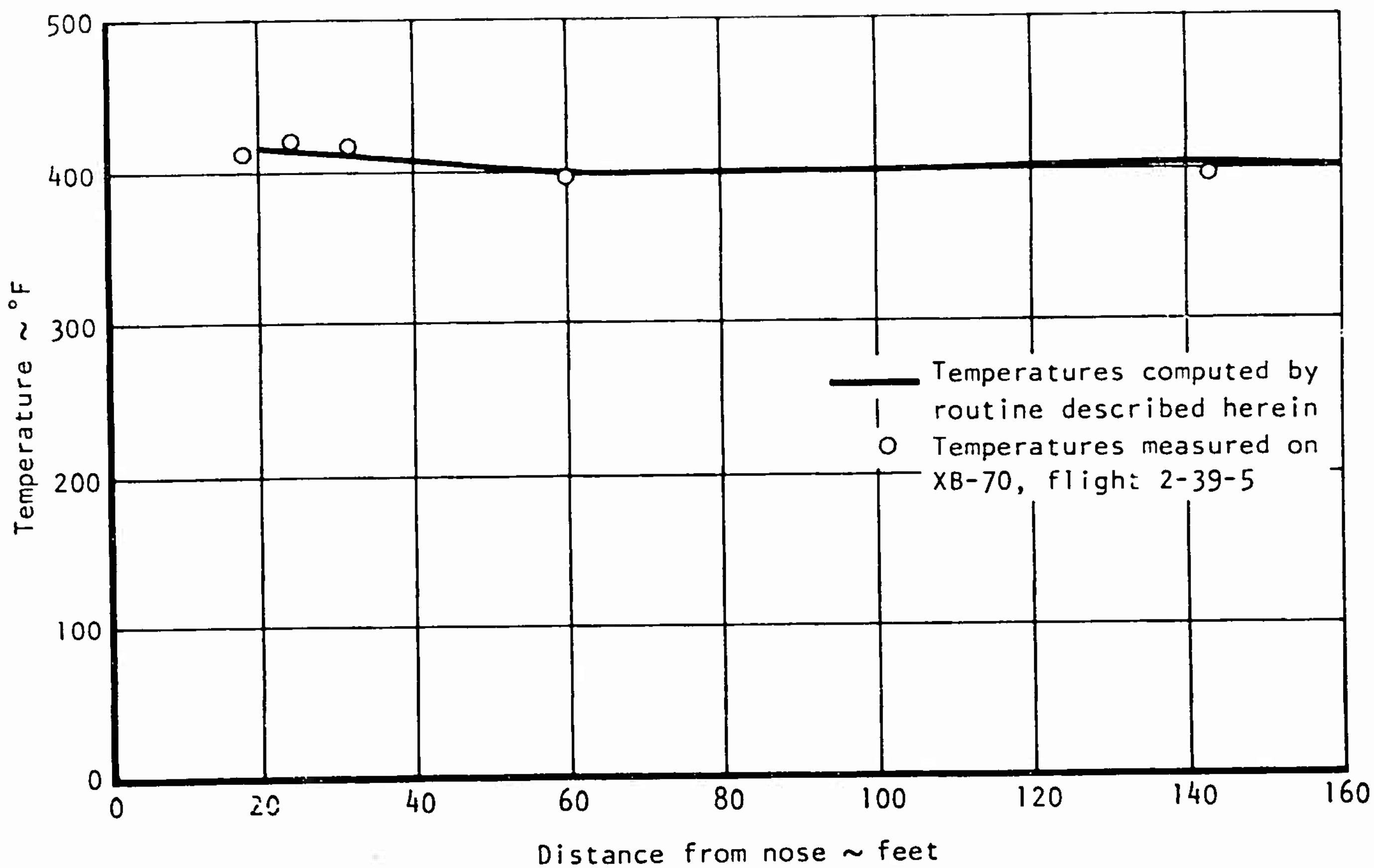



Figure 4. Comparison of computed and measured fuselage temperatures.

The most dominating heating or cooling effect in this program is the aerodynamic convection. The aerodynamic heating coefficient is computed by the well-recognized Blassius equation for laminar flow and the Van Driest method for turbulent flow. The latter will apply in most situations. Spalding and Chi[1] found it to be the best of 20 methods they checked with all available data. It checked almost as well as Spalding's own method based on these same data. Hopkins and Inouye[2] recommended it, based on a comparison of several leading methods. They suggest its application into the hypersonic regime.

The forcing potential for aerodynamic heating is based on the total temperature which, in turn, depends on specific heat. The specific heat as used here varies as a function of temperature in recognition of real gas effects, but it does not include effects of dissociation. Hence, it is quite accurate at subsonic and supersonic speeds, but it becomes progressively less accurate as one enters the hypersonic regime. For instance, at 2,240° F (1,500° K), this method gives 4.5° F error compared with the data of Reference 3, but at 3,140° F (2,000° K), the error has increased to 43.1° F. These temperatures correspond to flight at mach numbers 5.60 and 6.77, respectively.

## APPROACH TO FLUTTER ESTIMATION

### BACKGROUND

With the advent of higher flight speeds, the magnitude of flutter problems has become a major design consideration.

The requirements or design features necessary to prevent flutter often conflict with other design requirements. For instance, a thin, high-aspect-ratio wing may be desired for aerodynamic reasons. However, the weight increment over that resulting from strength requirements increases with increasing aspect ratio or decreasing thickness. The optimum wing then would probably be somewhat thicker and of lower aspect ratio than it would be if designed by strength and aerodynamic considerations only.

Unfortunately, it is impossible to formulate a true flutter analysis in which the required flutter speed is used as input to compute the optimum distribution of stiffness required. In fact, it is impossible to write an equation or set of equations for practical aircraft structures which can be solved directly for the flutter speed, even if all mass and stiffness properties are known. The reasons for this and methods for circumventing it are discussed in References 17 and 18. Briefly stated, the difficulty lies in the fact that the equations of motion are non-self-adjoint. The oscillatory aerodynamic forces are frequency and velocity-dependent, so that the speed and frequency at which flutter occurs must be known before the aerodynamic forces at flutter can be calculated.

The equations of motion are determined by the integration of many parameters over the entire structure. Thus, the distribution of quantities such as mass and stiffness are as important as the overall level, or the value at any particular point. In preliminary design stages, the level and distribution of these quantities are hardly ever known with sufficient accuracy to conduct a detailed flutter analysis. Therefore, it is evident that conducting an actual flutter analysis with detailed mass and stiffness distributions is not practical for preliminary design evaluations. Furthermore, sufficient time for such detailed analyses is usually not available.

SCOPE

Semiempirical methods have been developed for use in the preliminary design phase of vehicle synthesis. These methods are techniques of approximation that replace detailed flutter analyses. Techniques which have been incorporated in this program evaluate the prevention of:

1. Local panel flutter

2. Lifting surface flutter

3. T-tail flutter

Local panel flutter is generally associated with thin-skinned structures. This design consideration is evaluated for fuselage and nacelle structures. Discussion of the programmed methods are presented in Volumes V and VII of this document.

The development of the method used to evaluate surface flutter is based on the observation of some degree of correlation between the flutter speed and the static aeroelastic torsional divergence speed of an equivalent straight wing. The correlation is expressed by means of a parameter, $e_e$, which is a function of aspect ratio, sweep angle, and taper ratio. The expression for the parameter was derived from an envelope curve around a large number of points corresponding to theoretical and experimental evaluations of flutter speeds of a large variety of lifting surface types. The use of this "torsional divergence criteria" greatly simplifies the problem. The aerodynamic forces are not frequency-dependent, and inertia forces are eliminated from the problem. Naturally, some accuracy is lost, but since the correlation parameter is determined from actual flutter data, it may be thought of a "taking an average" of inertia and other effects to which torsional divergence considerations alone cannot relate.

The method determines the optimum stiffness distribution by calculating the stiffness distribution which will result in a constant shear stress over the span of the wing, in the torsional divergence mode.

The technique was originally developed to evaluate flutter at subsonic speeds. However, an extension has been developed for modifying the results to obtain stiffness predictions for flutter at transonic and supersonic speeds.

Derivation of the equations for calculating optimum stiffness distribution required to prevent flutter are described in Section II.

T-tails very often require considerably higher levels of stiffness to prevent flutter than do conventional tails. The stiffness increase above the levels required for strength is commonly higher in the vertical stabilizer, or fin, than in the horizontal stabilizer. Consequently, in preliminary design studies of T-tail aircraft it is necessary to make some estimate of the requirements for flutter prevention in order to have confidence that the tail design is realistic. It may be impossible to achieve a design which is actually optimum if flutter is not considered. In fact, it is conceivable that the stiffness requirements for a T-tail on a high-speed aircraft could be so high that the T-tail configuration would not represent an optimum design for the particular aircraft performance requirements. This fact may not be realized early enough in the design cycle to allow alternates to the T-tail to be selected, if preliminary estimates of the flutter prevention requirements are not made.

The programmed method is based on scaling from results of Reference 19 to establish required stiffness level. The correlation parameter, $e_e$, is used to account for differences in planform geometry.

## Section II

## METHODS AND FORMULATIONS

### TEMPERATURE METHODOLOGY

### HEAT BALANCE

The skin temperatures are solved from a heat balance in which the sum of five terms is equated to zero. The heat balance is merely a statement of the law of conservation of energy as applied to an element of skin. (See Figure 5.) In words, this balance is:

Zero = aerodynamic heat (1)
+ solar heat
+ radiation between skin and ambient
+ heat from sundry sources
+ heat conducted or convected from interior

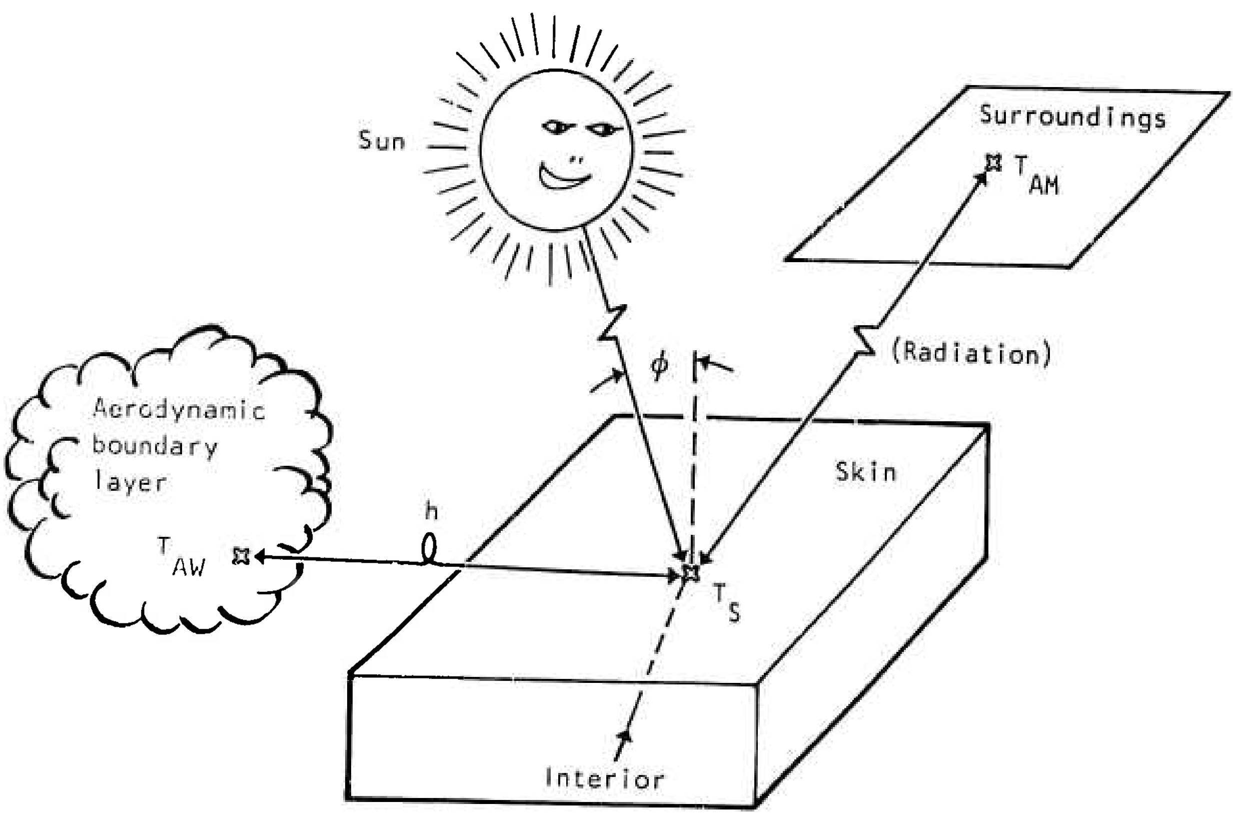



Figure 5. Heat balance diagram

This equation follows the convention of considering all terms as positive; i.e., all terms are written to express heat flowing into the skin. Under this convention, a term will become negative automatically when the skin's temperature exceeds the temperature of the connected node.

Presently, this program computes what is known as the "radiation equilibrium temperature" by setting the last two terms to zero. They are terms which express heat from within the airplane, and frequently they are not applicable. Large sections of an airplane closely approach the radiation equilibrium temperature. Examples are wing and tail surfaces which are externally heated from both sides or the cabin's structure where insulation prevents heat from flowing from the interior.

## Flux Equation

All these heat transfer terms are based on the same area and time interval. These are divided out to change the equation, as it appears in the program, from an energy balance to a flux balance. Each term is now in BTU per hour per square foot. Equation 2, the flux equation, is shown for the aerodynamic heat, solar heat, and radiation between skin and ambient heat transfer terms.

$$Q = h\,(T_{AW} - T_S) + \alpha G \cos\phi + \sigma \epsilon\,(T_{AM}^4 - T_S^4) \qquad (2)$$

where

$h$ = aerodynamic heat transfer coefficient, BTU/hr/°R/ft$^2$

$T_{AW}$ = adiabatic wall temperature, °R

$T_S$ = skin temperature, °R

$\alpha$ = absorptivity of surface to solar irradiation, (no units)

$G$ = solar flux through a plane perpendicular to sun beams, BTU/hr/ft$^2$

$\phi$ = solar angle of incidence measured from normal to surface, degrees ($\phi$ is assumed to be zero in this program)

$\sigma$ = Steffan-Boltzman constant, 1714 x $10^{-12}$ BTU/hr/°R$^4$/ft$^2$

$\epsilon$ = emittance of skin at skin temperature (no units)

$T_{AM}$ = mean ambient temperature, °R

Derivation of each of these three heat transfer terms is discussed in the order that they are determined in this program.

## SOLAR HEAT FLUX

The solar heating flux term in equation 2 is:

$$\alpha G \cos\phi = \text{solar heat flux, BTU/hr/ft}^2 \quad (3)$$

### Solar Flux

Sunshine is highly variant as evidenced by the spread of lines in figure 6. It varies with cloud cover, time of day, and solar activity. A rigorous standard for values to be used for aircraft design does not exist except for sea level. Reference 14, which has semihandbook status, recommends the curve shown in Figure 6. The zero altitude value corresponds to the value specified for military equipment on the ground by Reference 12. The asymptote of 435 $BTU/hr/ft^2$ is a frequently used value for the flux in the upper air above almost all of the water vapor, carbon dioxide, and dust that filter sunlight in the lower reaches of the atmosphere.

A curve fit of this heavy line is contained in a small function subprogram SOLARG. The program calls this function using altitude as a parameter.

### Absorptivity and Emissivity

The solar absorptivity, $\alpha$, and the emissivity, $\epsilon$, which appears in the radiation between skin and ambient heat flux term, should not be confused. These terms are independent of each other. Both depend upon the paint, plating, anodizing, or other surface treatment that the airplane may receive. This independence evolves from the different source temperatures in these two radiation terms. The sun is about 10,000° F, and its radiation is in the short-wave portion of the spectrum. The radiation between the skin and the ambient is in the far infrared portion of the spectrum, as the sources are relatively cool. Between these two portions, there is ample room for changes in radiative properties. For example, a reduction of 70° F in skin temperature was obtained on the B-70 by using a finish which was both highly reflective in the solar portion of the spectrum and highly emissive in the far infrared portion. Absorptivity of 0.85 is used in this program. This value is representative of many painted surfaces. Should actual absorptivity be available, the user may input the value for this term.

The angle of incidence, $\phi$, is presently taken as zero; i.e., the sunbeams strike perpendicularly. This maximizes the solar heat flux entering the skin, a permissible simplification in preliminary design. Angle of incidence may also be varied by the user.

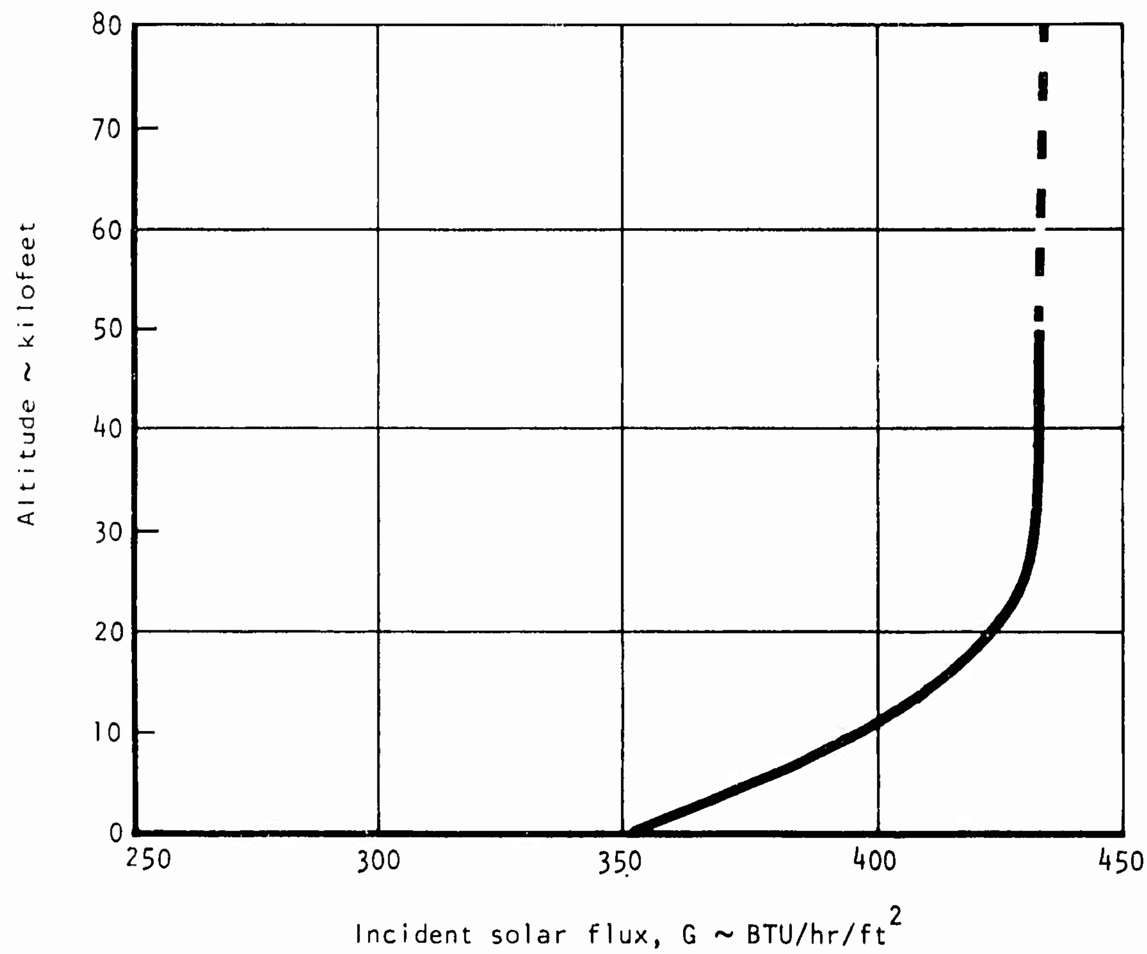



Figure 6. Recommended curve of maximum solar irradiation.

RADIATION FLUX BETWEEN SKIN AND AMBIENT

Radiation between skin and ambient can be regarded as the radiant cooling term, for the net radiation is usually away from the skin. It is written as a positive term in accordance with the convention of signs, and it automatically becomes negative when the skin's temperature exceeds the ambient temperature. The term is written as

$$\sigma \epsilon \left( T_{AM}^{4} - T_{S}^{4} \right) \qquad (4)$$

where

$\sigma$ = Steffan-Boltzman constant

$= 1.714 \times 10^{-12}$ BTU/ft$^2$/hr/°R$^4$

$\epsilon$ = emittance of skin at skin's temperature (no units)

$T_{AM}$ = mean ambient temperature, °R

$T_S$ = skin temperature, °R

Emissivity of 0.85 is used in this program. Should actual structure emissivity be known, the user may input the value for this term.

## Nocturnal Radiation

The mean ambient temperature in equation 4, $T_{AM}$, is the fourth root of the sum of the products of the fourth powers of all surrounding temperatures (i.e., clouds, ground, atmosphere, etc) and their associated view factors from the skin. Its corresponding radiation is referred to as "nocturnal radiation."

Nocturnal radiation, the radiation from earth and sky, is usually a small term, and it is ignored; i.e., $T_{AM}$ is set arbitrarily to zero. This partially balances the assumption that the sun shines perpendicular to the skin, a possible but not a probable occurrence. In the upper reaches of the atmosphere, nocturnal radiation approaches zero and, at low altitude, it is usually small compared with the aerodynamic heating term. Table 5, containing data extrapolated from Reference 15, shows the magnitudes. The data shown for the warm condition are the average of two sets of data - one for a clear, hot, moist night at Phoenix, Arizona, and the other for a clear, mild night at Dallas, Texas. Data shown for the cool condition are for a clear, cold, and dry night at Barrow, Alaska. The temperatures shown are those of a black-body that would produce the same radiant flux on the surface.

TABLE 5. NOCTURNAL RADIATIONS AND EQUIVALENT BLACK-BODY TEMPERATURES

| Altitude (feet) | Warm Condition | | | | Cool Condition | | | |
|---|---|---|---|---|---|---|---|---|
| | From Above | | From Below | | From Above | | From Below | |
| | Flux (*) | Temp (°R) | Flux (*) | Temp (°R) | Flux (*) | Temp (°R) | Flux (*) | Temp (°R) |
| 0 | 101 | 493 | 139 | 533 | 37 | 383 | 67 | 445 |
| 1,000 | 97 | 487 | 137 | 531 | 37 | 383 | 68 | 447 |
| 5,000 | 76 | 460 | 127 | 522 | 34 | 375 | 65 | 442 |
| 10,000 | 57 | 427 | 113 | 506 | 27 | 355 | 56 | 427 |
| 20,000 | 35 | 386 | 91 | 481 | 18 | 320 | 39 | 389 |
| 30,000 | 23 | 339 | 78 | 462 | 14 | 298 | 29 | 361 |
| 40,000 | 16 | 310 | 68 | 446 | 10 | 275 | 23 | 338 |
| 50,000 | 11 | 285 | 60 | 432 | 8 | 257 | 18 | 320 |
| 60,000 | 8 | 263 | 53 | 419 | 6 | 244 | 15 | 307 |
| 70,000 | 6 | 243 | 47 | 408 | 5 | 230 | 12 | 292 |
| 80,000 | 4 | 225 | 42 | 397 | 4 | 218 | 10 | 280 |
| 90,000 | 3 | 207 | 38 | 386 | 3 | 204 | 9 | 268 |
| 100,000 | 2 | 191 | 34 | 375 | 2 | 185 | 7 | 255 |

*BTU/hr/sq ft

AERODYNAMIC HEAT FLUX

The aerodynamic heat flux term in equation 2 is:

$$h(T_{AW} - T_s) = \text{aerodynamic heat flux, BTU/hr/ft}^2 \quad (5)$$

Atmospheric Properties

Aerodynamic heat transfer coefficient, h, and adiabatic wall temperature, $T_{AW}$, are dependent on vehicle speed and atmospheric properties at the specific flight condition. Function routines PRESH, TEMALT, and TTO develop this atmospheric data.

Local Static Pressure

The local static pressure is assumed equal to the free-stream static pressure; i.e., the pressure coefficients are assumed zero. This assumption

is generally valid for large flat sections alined with the flow. At particular locations such as leading edges, canopies, windshields, etc, it causes error. Figure 7 illustrates the relative error for a delta wing at mach 3.00 and 65,000 feet. Here the differences of local pressure are holding the airplane aloft. The two solid curves were computed using the correct local static pressures and the local static temperatures. The dotted curve was computed from the free-stream values.

The free-stream static pressure is computed by the function subprogram PRESH, for "pressure height." This function is entered with the pressure altitude. It uses the hydrostatic equations and temperature distribution of the standard atmosphere (11).

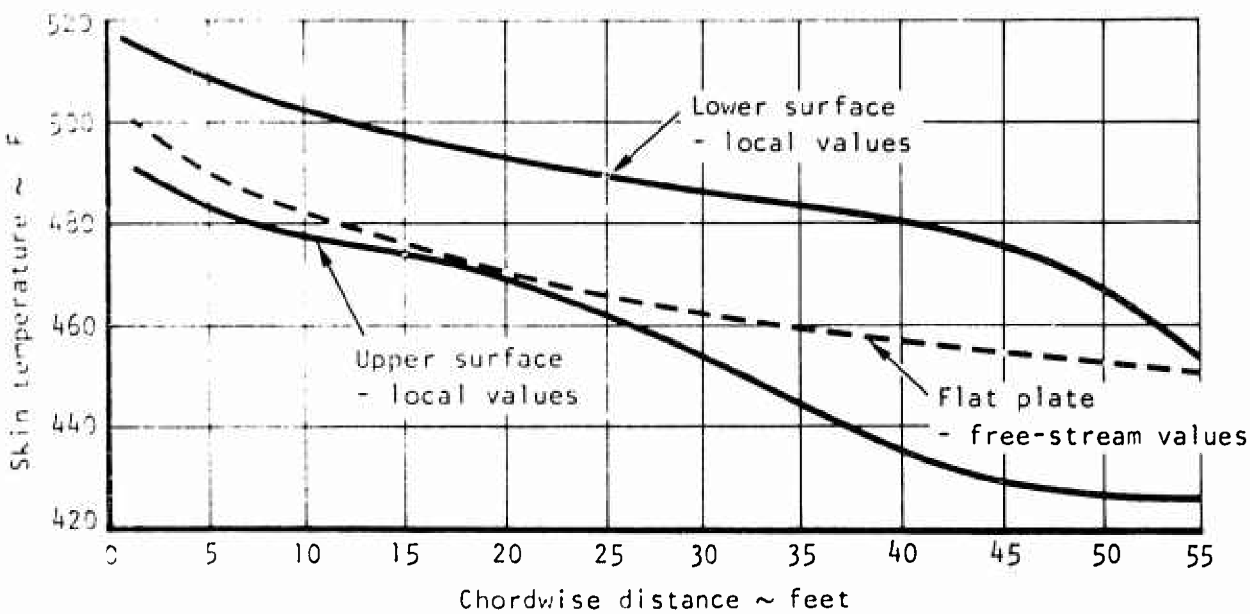



Figure 7. Comparison of using local and free-stream properties for a large wing at 3.00 mach and 65,000 feet.

Local Static Temperature

The local static temperature is used in the three aerodynamic heating subprograms: HBL, TBL, and TT0. It is assumed equal to the free-stream static temperature. Local static temperatures vary less percentagewise from free-stream static temperatures than the local pressure does from the free-stream pressure, as previously discussed. Hence, this assumption is justified for preliminary design.

The free-stream static temperature is computed from the pressure altitude by the function subprogram TEMALT, for "temperature versus altitude." This function consists of a series of usually linear equations which represent the segments of lines of Figure 8. The function contains six model atmospheres,

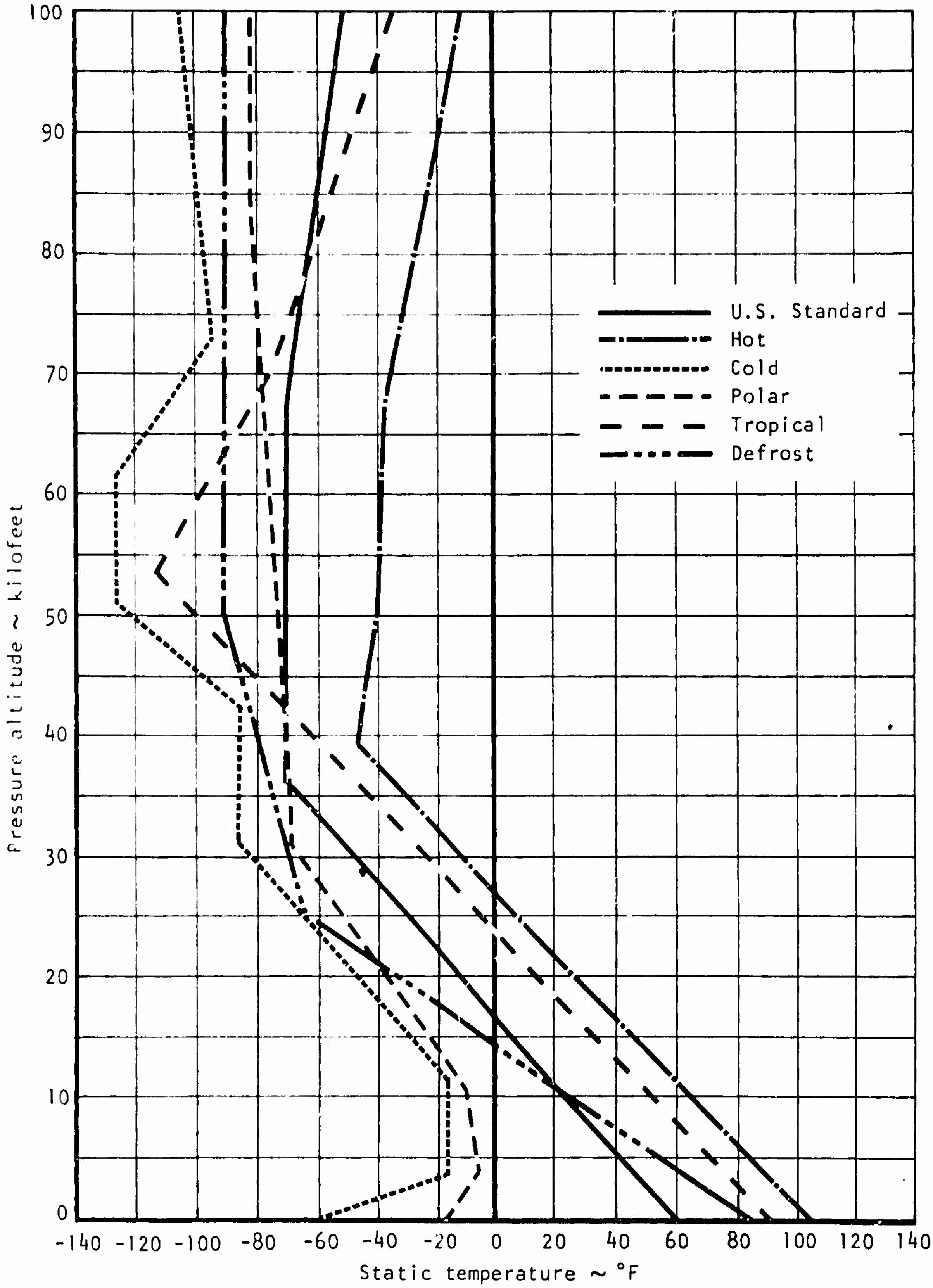



Figure 8. Static temperatures of model atmospheres.

of which only the standard atmosphere is currently used. These atmospheres and the indexing numbers for calling them are listed in Table 6.

TABLE 6. CALLING NUMBERS FOR MODEL ATMOSPHERES

| Call | Model Atmosphere | Source Ref |
|---|---|---|
| 1 | U.S. Standard Atmosphere, 1962 | 11 |
| 2 | Cold Atmosphere (MIL-STD-210A) | 12 |
| 3 | Hot Atmosphere (MIL-STD-210A) | 12 |
| 4 | Polar Atmosphere (MIL-STD-210A) | 12 |
| 5 | Tropical Atmosphere (MIL-STD-210A) | 12 |
| 6 | Defrost Atmosphere | 13 |

Total Temperature

Both the function for the aerodynamic heating coefficient and the function for the adiabatic wall temperature require the total temperature internally.

The total temperature is defined as the temperature reached by bringing air to rest adiabatically and with no addition or subtraction of work. This definition is used to compute the total temperature by the method of constant total enthalpy where total enthalpy is defined as the sum of the static enthalpy and the kinetic energy.

$$H_T = H_1 + KE_1 = H_2 + KE_2 \tag{6}$$

where $H$ = enthalpy

$KE$ = kinetic energy

Subscripts T, 1, and 2 = total, initial, and final, respectively

By definition, the final kinetic energy is zero. Then:

$$KE_1 = H_2 - H_1 = \int_{T_1}^{T_2} wc_p \, dT = \frac{w V_1^2}{2gJ} \tag{7}$$

where

$w$ = weight of air, lb

$c_p$ = specific heat of air, BTU/lb/°R

$g$ = gravitational acceleration, ft/sec$^2$

$J$ = 778 ft-lb/BTU = Joule's constant

$V_1$ = initial local velocity, ft/sec

Specific heat varies as a function of temperature according to an equation by Donaldson (8).

$$c_p = \frac{R}{J}\left[\frac{7}{2} + \left(\frac{\theta/T}{e^{\theta/T} - 1.000}\right)^2 e^{\theta/T}\right] \tag{8}$$

where

$c_p$ = specific heat of air, BTU/lb/°R

$R$ = gas constant of air = 53.35 ft/°R

$\theta$ = 5,526° R = characteristic temperature of molecular vibration

$T$ = varying static temperature of air as it is brought to rest, °R

The variable specific heat, equation 8, is substituted into equation 7, the weight cancelled from both terms, and the integration made between the limits of the static temperature, $T_L$, and the total temperature $T_T$.

$$\frac{V^2}{2gJ} = \frac{R}{J}\left[\frac{7}{2}\left(T_T - T_L\right) + \frac{\theta}{e^{\theta/T_T} - 1.000} - \frac{\theta}{e^{\theta/T_L} - 1.000}\right] \tag{9}$$

Equation 9 is solved by a trial and error process to return the total temperature, $T_T$.

## Adiabatic Wall Temperature

The adiabatic wall temperature is the forcing potential, a mean effective temperature for heat transfer through the aerodynamic boundary layer. The adiabatic wall temperature, $T_{AW}$, is computed in the function subprogram, TBL. The equation for adiabatic wall temperature is:

$$T_{AW} = T_L + r\left(T_T - T_L\right) \qquad (10)$$

where

$T_{AW}$ = adiabatic wall temperature, °R

$T_L$ = local static temperature, °R

$r$ = recovery factor

$T_T$ = total temperature, °R

The recovery factor is the square root of the Prandtl number for laminar flow and the cube root for turbulent flow. The free-stream Reynolds number is compared with the criterion for transition, and the square root or the cube root is selected accordingly. The Prandtl number varies with the reference temperature.

## Reference Temperature

The thermal properties of air in both the HBL and TBL function subprograms are evaluated at the reference temperature as defined by Eckert[5].

$$T_{ref} = 0.280\ T_L + 0.500\ T_S + 0.220\ T_{AW} \qquad (11)$$

where

$T_{ref}$ = reference temperature, °R

$T_L$ = local static temperature, °R

$T_S$ = skin temperature, °R

$T_{AW}$ = adiabatic wall temperature, °R

This reference temperature may be regarded as a weighted average of the temperatures that exist at various locations through the boundary layer.

Conductivity, Viscosity, Prandtl Number, and Reynolds Number

Thermal conductivity, viscosity, Prandtl number, and Reynolds number are used in the function subprograms for aerodynamic heating coefficient, h, and adiabatic wall temperature, $T_{AW}$. These properties vary according to the reference temperature which depends on the adiabatic wall temperature. These functions have trial and error solutions to converge iteratively on their answers.

Variations of conductivity and viscosity are calculated as recommended in References 10 and 11.

$$k = \frac{0.0011404\, T_{ref}^{1/2}}{1 + \frac{441.72}{T_{ref}} (10.00)^{21.6/T_{ref}}} \tag{12}$$

$$g\mu = 0.002627 \frac{T_{ref}^{1.5}}{T_{ref} + 198.7} \tag{13}$$

where

$k$ = thermal conductivity, BTU ft/hr/°F/ft$^2$

$g\mu$ = viscosity times gravitational acceleration, lb/hr/ft

$T_{ref}$ = reference temperature, °R

$$N_{PR} = \text{Prandtl number, } \frac{c_p \mu g}{k} \tag{14}$$

$$N_{RE} = \text{Reynolds number, } \frac{\rho V L}{\mu} \tag{15}$$

where

V = Local flow-velocity, ft/sec

L = Characteristics-Length, taken as half the surface mean geometric chord

$c_p$ = specific heat, BTU/lb/°R (equation 7)

## Aerodynamic Heating Coefficient

The aerodynamic heating coefficient, h, is computed by a function subprogram, HBL. This subprogram was written to calculate h at one specific point on any of the following categories of surfaces:

1. Flat plates at zero angle of attack (parallel flow). Parameter h is calculated at the specified distance aft of the leading edge.

2. Flow over conical surfaces with body centerline at zero angle of attack. Parameter h is calculated at the specified distance aft of the leading edge.

3. Swept cylindrical leading edges. Parameter h is calculated at the leading edge.

4. Spherical bodies. Parameter h is calculated at the stagnation point.

Only method 1, specified in the calling program TEMPER, is used by SWEEP, since temperatures chosen for the structural design of all surfaces are those occurring at a distance 10 feet aft of the leading edge of a flat plate.

Subprogram HBL was written to account for either laminar or turbulent flow, depending upon whether the point at which the structural temperature is to be calculated lies forward or aft of the flow transition point. When the input value at transition Reynolds number is zero, as currently programmed, subprogram HBL automatically selects 1 million for that value. With the combination of 1 million for transition Reynolds number and 10.0 feet for the point of temperature calculation, the program will, in virtually all cases, determine h based upon turbulent flow properties.

## Transition

The flow in the parallel and conical orientations varies automatically from laminar to turbulent and vice versa according to the preset criterion for change. The program compares the actual Reynolds number with this criterion and uses the appropriate mode of flow. Entering a very high Reynolds number of transition will cause only laminar flow to be considered. The default for this criterion is one million, a frequently used but not necessarily accurate value. Truitt[4] says, "The present state of the art unfortunately does not allow one to predict accurately the transition on a given body. Many factors must be taken into account and properly weighed. Except for the stagnation region problem, it is probably better for a preliminary analysis to assume that turbulent flow exists above local Reynolds number of one million." The precise value of the transitional criterion is not usually important, as transition generally occurs near the leading edge causing most of the airplane to be in the turbulent regime.

## Laminar Coefficient

The laminar heat transfer coefficient is computed from the familiar Blassius equation

$$C_f = 0.664\, N_{RE}^{-0.500} \tag{16}$$

where

$C_f$ = skin friction coefficient

$N_{RE}$ = Reynolds number

This skin friction coefficient is transposed to a heat transfer coefficient by Reynolds analogy modified as recommended by Eckert[5].

$$h = \frac{C_f}{2}\, N_{PR}^{-2/3}\, g\rho c_p\, V \tag{17}$$

where

$h$ = heat transfer coefficient, BTU/hr/°R/ft$^2$

$C_f$ = skin friction coefficient, no units

$N_{PR}$ = Prandtl number

$g\rho$ = specific weight, lb/ft$^3$

$c_p$ = specific heat, BTU/lb/°R

$V$ = local velocity, ft/sec

Turbulent Coefficient

For turbulent flow, the program calculates heat transfer coefficient by the Van Driest method. Actual detail and derivation of the equations used by Van Driest are found in References 6 and 7. Truitt [4] presents the equations in a workable form for engineering calculations.

## SOLUTION FOR SKIN TEMPERATURE

The heat flux equation, based on the assumption that noctural radiation is negligible, can be written as shown in equation 18.

$$0 = h\left(T_{AW} - T_S\right) + \alpha G \cos\phi - \sigma\epsilon T_S^4 \qquad (18)$$

Equation 18 can be rearranged as shown in equation 19.

$$T_S = A - B\, T_S^4 \qquad (19)$$

where

$$A = \frac{h\, T_{AW} + \alpha G \cos\phi}{h}$$

$$B = \frac{\sigma\epsilon}{h}$$

Equation 19 is an inverted fourth-power parabola which is solved by a trial and error method that returns the only real and positive root. The method is a modified chord or reguli falsi method. The solution is performed by the function TSKIN, for "temperature of skin." It is described in detail by Reference 16.

Since both adiabatic wall temperature, $T_{AW}$, and aerodynamic heat transfer coefficient, h, are dependent on the equilibrium skin temperature, an iteration procedure is used. Subroutine TEMPER controls this iteration. The search loop is outlined as follows:

1. Estimate skin temperature.

2. Use TBL to determine adiabatic wall temperature based on the estimated skin temperature, equation 10.

3. Use HBL to determine aerodynamic heat transfer coefficient based on the estimated skin temperature.

4. Use TSKIN to calculate skin temperature, equation 19.

5. Test on estimated versus calculated temperature (Step 1 versus Step 4).

6. If calculated temperature has not converged sufficiently, enter loop at Step 2 and repeat using calculated for estimated skin temperature.

The program contains a criterion of convergency to which the difference of answers from successive passes are compared. One-tenth of a degree Fahrenheit (0.1° F) is used in the present version of the program. If the criterion is not met within 90 passes, the convergence criterion is raised to 1 degree. Then, up to 10 additional passes are allowed to attempt to converge, but in any case the final values are used.

## MATERIALS PROPERTIES METHODOLOGY

For a discussion of the methodolgy used in determining materials properties, refer to Part 1 of this volume.

## PANEL FLUTTER METHODOLOGY

Since panel flutter is evaluated only for fuselage and nacelle structures, discussion of this methodology is presented in the appropriate sections of Volumes V and VII.

## LIFTING SURFACE FLUTTER METHODOLOGY

In a detailed flutter analysis, aerodynamic coefficients are determined for several values of the dimensionless parameter $b\omega/V$ (characteristic length times frequency of oscillation, divided by velocity). The equations of motion may then be written in the form of complex eigenvalue problem and solved at each value of $b\omega/V$ for frequency, velocity, and structural damping required to sustain harmonic oscillations. The damping required for harmonic oscillation is plotted versus velocity. The point on the curve at which damping required is equal to the damping which exists in the structure is the neutral stability point.

A truly optimum stiffness distribution would be one which resulted in a constant stress over the entire structure if it were deformed in the flutter mode. The flutter mode is determined by the combination of the aerodynamic and the inertia loading. The maximum loading is determined by the vectorial sum of the aerodynamic and inertia loads, since phase angles exist between the two types of load. However, according to Reference 18, examination of several flutter cases has shown that, for the planforms considered, the two types of loads are distributed very nearly in the same manner. Hence, a consideration of the aerodynamic loading alone provides a good approximation of the loading. Furthermore, it has been noted that the aerodynamic loading at the flutter point very closely resembles the loading at static aeroelastic, torsional divergence of an equivalent straight wing. This provides a very powerful tool for development of the technique for predicting stiffness required to prevent flutter. An important implication is that only the torsion stiffness need be considered. Naturally, this dictates that cases in which the ratio of bending frequency to torsion frequency is too close to unity are, in general, beyond the scope of the technique. These cases, however, are sufficiently rare that the majority of cases may be evaluated.

## DERIVATION OF EQUATIONS

The dynamic pressure at torsional divergence of an unswept wing is given by the equation

$$q = \frac{\int GJ\theta'^2 dx}{C_{\ell_\alpha} e \int C^2 \theta^2 dx} \qquad (20)$$

From this, a stiffness criterion has been defined in Reference 20.

$$C_{\ell_\alpha} eq = \frac{\int GJ\,\theta'^2 dx}{\int c^2\,\theta^2 dx} \tag{21}$$

For convenience, the product $C_{\ell_\alpha} eq$ has been defined arbitrarily as

$$e_e = \left(\frac{V_a}{1000}\right)^2 \frac{C_{\ell_\alpha} e_1}{2\pi} \quad (e_1 \text{ in \% chord}) \tag{22}$$

or

$$e_e = C_{\ell_\alpha} e_1 q/1.116 \quad (q \text{ in psi})$$

The parameter $e_e$ is called the "effective eccentricity" of the surface, and can be thought of as the chordwise eccentricity the structure can support at divergence at 1,000 mph if the lift curve slope is equal to $2\pi$.

Examination of a large collection of experimental flutter model data has shown that the parameter $e_e$ is primarily a function of aspect ratio and sweepback angle. An empirical expression has been developed which gives the proper value of $e_e$ to correlate well with a large amount of available data. This expression is, for the foregoing definition of $e_e$,

$$e_e = \frac{K}{(1 + 0.8/AR)^2} \frac{q}{1480} [0.4 + 0.7 \cos (\Lambda_{c/4} - 10°)] \tag{23}$$

where

K = 24.75 for wings

and

K = 25.88 for empennage surfaces.

This expression has been used with considerable success for the speed ranges below those where appreciable compressibility effects are encountered. A modification to be used for transonic and supersonic design points has been proposed and will be discussed in a later paragraph.

A technique has been developed for computing the torsional stiffness distribution which will result in a constant stress over the span of the wing for the torsional divergence mode. This technique will be presented in the matrix form. The wing is divided into chordwise strips as shown in Figure 9. The strips of width, $\Delta x_i$, are used for the definition of the aerodynamic terms, while the strips of width, $\Delta X_K$, are used for definition of structural properties. The torsional stiffness of the wing is defined by the torsional constant, $J_K$, at the outboard edge of the $K_{th}$ strip. The relation between twist angle and torsional loading

$$\frac{d\theta}{dx} = \frac{T}{GJ} \tag{24}$$

may be written in matrix form for the wing of Figure 9 as

$$G \left[ J_{kk} \right] \{ \theta_i' \} = \{ T_i \} \tag{25}$$

where $\theta_i'$ is interpreted as the change in twist angle over the $i^{th}$ strip and $T_i$ is the total torque on the structure at the $i^{th}$ strip. Since the total torque at any station is a summation of the individual torque acting on the strip and on each strip outboard of the strip considered,

$$\{ T_i \} = \left[ I_n \right]^T \{ \Delta T_i \} \tag{26}$$

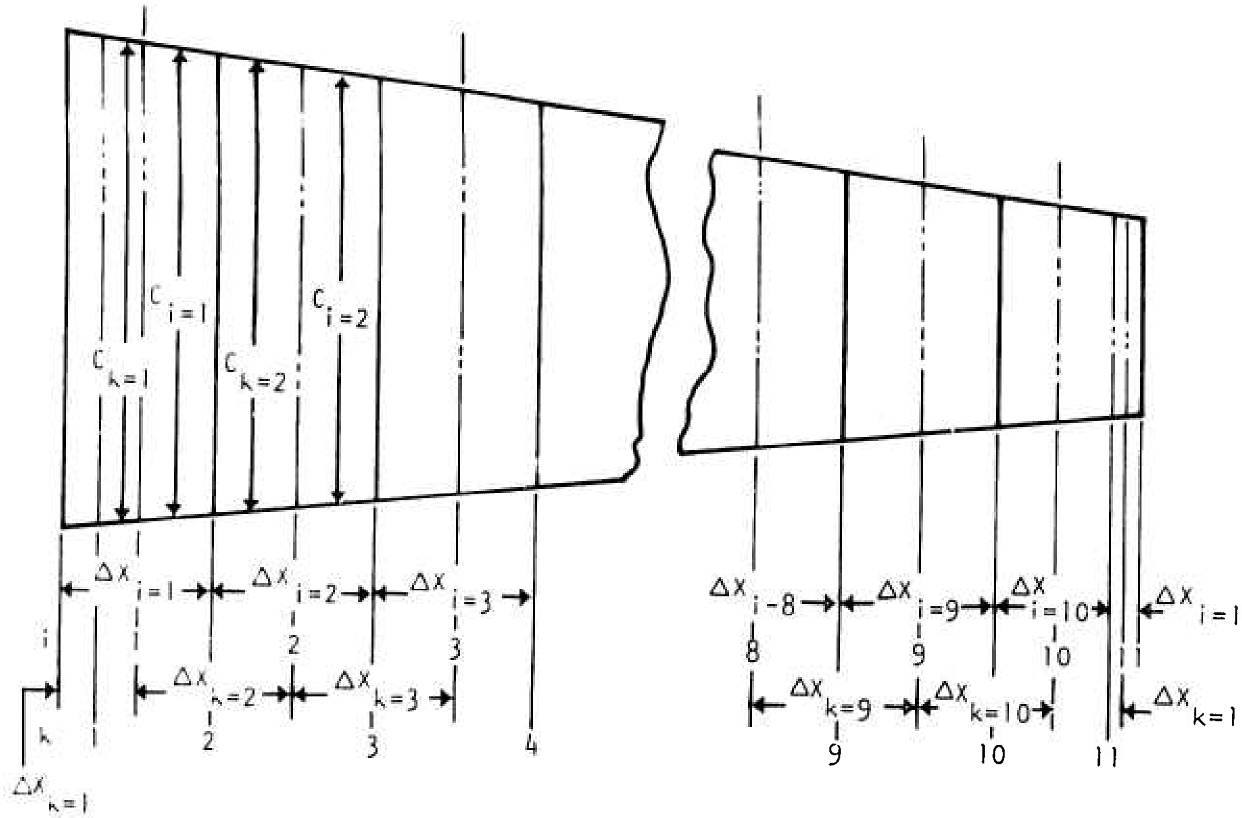



Figure 9. Wing geometry for derivation.

where

$$[I_n]^T = \begin{bmatrix} 1 & 1 & - & - & - & 1 \\ 0 & 1 & 1 & - & - & 1 \\ \cdot & \cdot & 1 & - & - & \cdot \\ \cdot & \cdot & \cdot & 1 & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & 1 & \cdot \\ 0 & \cdot & \cdot & \cdot & 0 & 1 \end{bmatrix} \tag{27}$$

and $\{\Delta T_i\}$ can be expressed as

$$\{\Delta T_i\} = C_{\ell\alpha} qe \, \lceil C^2 \rfloor \, \lceil \Delta X_i \rfloor \, \{\theta_i\} \tag{28}$$

The term $C_{\omega}qe$ has been defined in Reference 18 as 1.116 $e_e$. Using this notation,

$$G\lfloor J_{KK} \rfloor \{\theta'_i\} = 1.116\ e_e\ [I_n]^T \left\lfloor C_i^2 \right\rfloor \lfloor \Delta X_i \rfloor \{\theta_i\} \tag{29}$$

The condition that the torsional stress must be constant over the span requires that

$$\{T_i\} = 2\sigma_T \lfloor A_k \rfloor \{t_k\} \tag{30}$$

Also, the torsion constant $J_k$ can be defined as

$$\{J_k\} = \left\lfloor 4\ A_k^2 \right\rfloor \lfloor 1/S_k \rfloor \{t_k\} \tag{31}$$

Equation 31 may be rearranged to express the skin thicknesses as

$$\{t_i\} = 1/4 \left\lfloor 1/A_k^2 \right\rfloor \lfloor S_k \rfloor \{J_k\} \tag{32}$$

Placing equation 32 into equation 30 results in

$$\{T_i\} = 2\sigma_T/4 \lfloor S_k/A_k \rfloor \{J_k\} \tag{33}$$

Substituting equation 33 for $T_i$ in equation 25 and rearranging results in

$$\{\theta'_i\} = 2\sigma_T/4G \{S_k/A_k\} \tag{34}$$

A numerical integration of equation 34 may be expressed as

$$\{\theta_i\} = 2\sigma_T/4G [I_n] \lfloor \Delta X_{kk} \rfloor \{S_k/A_k\} \tag{35}$$

Equations 34 and 35 may now be substituted into equation 10 to obtain the expression for the required optimum stiffness distribution

$$\{GJ\} = 1.116\ e_e \left[\frac{A_k}{S_k}\right] \lfloor I_n \rfloor^{-1} \left[ C_i^2 \right] \lceil \Delta X_i \rfloor \lfloor I_n \rfloor \lceil \Delta X_k \rfloor \left\{ \frac{S_k}{A_k} \right\} \tag{36}$$

The values for $A_k/S_k$ may not be known or it may not be desirable to spend time computing these quantities for preliminary design approximations. However, there is a substitution which can be made for $A_k/S_k$ in most cases. If the width of the torque box is proportional to the wing chord and the thickness-to-chord ratio is constant across the span, then $A_k/S_k$ is proportional to the wing chord, for a rectangular torque box.

The advantage of this substitution is readily seen; the wing chord at the $K_{th}$ stations can be determined much more quickly than the value of $A_k/S_k$. The error due to this approximation is almost negligible, even in many practical cases where the preceding assumptions are not exactly true. This is primarily because the level of $A_k/S_k$ has no effect on the torsion constants computed. The values for $A_k/S_k$ are used once and the reciprocals are used once in the matrix equation. Thus, it is evident that if the column matrix $A_k/S_k$ were multiplied by any arbitrary constant and the column $\{S_k/A_k\}$ by the reciprocal of the constant, the net result would be no change in the equation. It may be concluded that only the relative distribution of values of $A_k/S_k$ across the span is important. Although most torque box cross sections are not exactly rectangular, their deviation from this shape does not in itself appreciably affect the manner in which the values of $A_k/S_k$ are distributed over the span.

An integrated form of equation 36 is presented in Reference 21. This form of the equation has been programmed in subroutines GJCAL and GJSI of the wing and empennage weight estimation module.

## APPLICATION TO SWEPTBACK WINGS

Although the previous derivation used a straight wing as the example planform, sweptback wings can also be handled. The relation for $c_e$, given by Equation 24 contains an empirical expression to account for sweepback angle. The only additional consideration necessary is that of defining the length and position of the elastic axis. Although no definite theoretical method has been formulated for this, techniques have been developed which have consistently given good results.

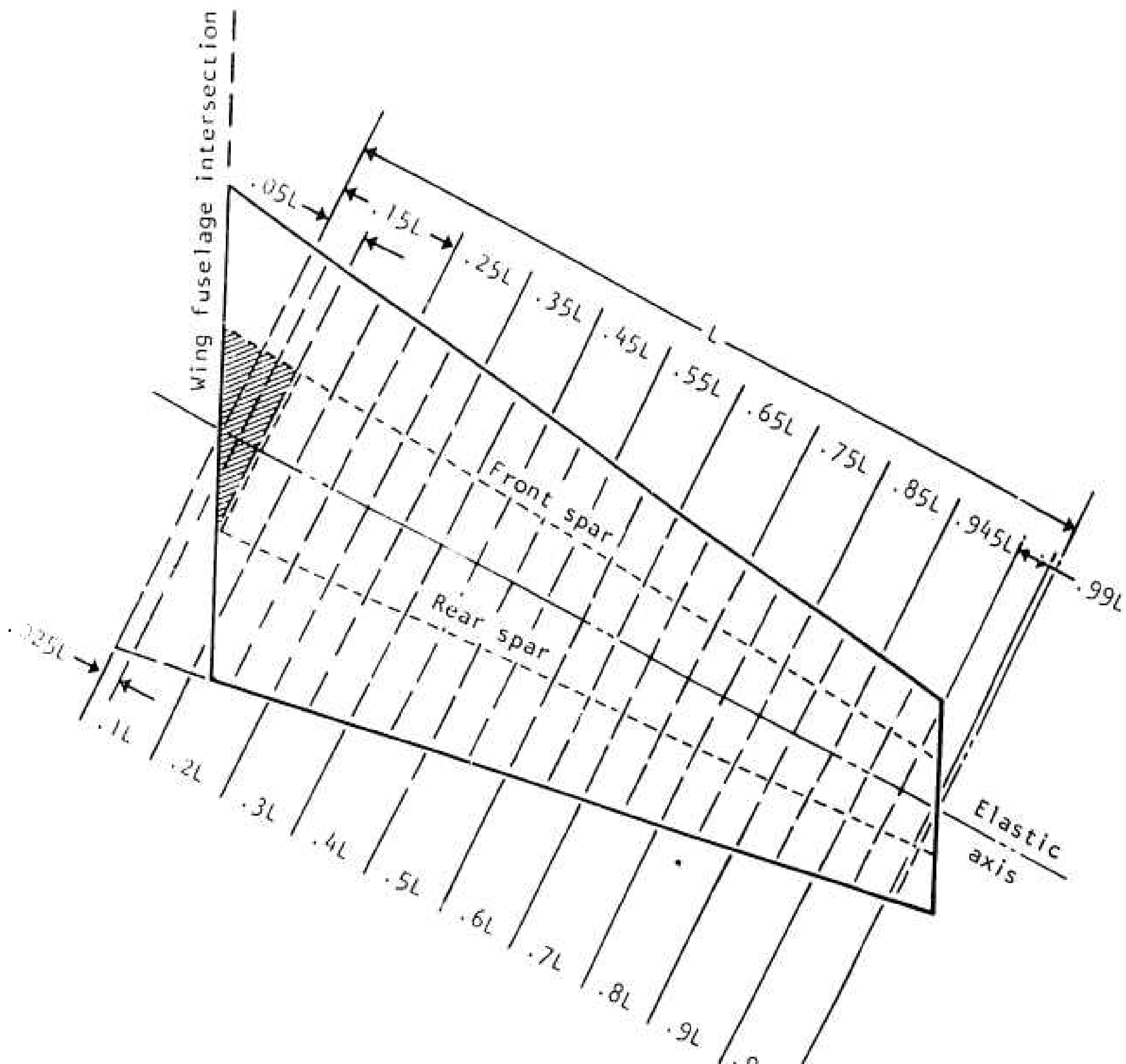
The geometry for a typical swept wing is shown in Figure 10. It has been found that best results are obtained if the chords perpendicular to the elastic axis are defined as shown in Figure 9. That is, the trailing edge is extended inboard of the wing-fuselage intersection and the chords in the wing root area are defined by the full distance between the leading edge and the extended trailing edge along the line normal to the elastic axis. The tip area, however, is not handled in a similar manner. The chords, measured along the normal to the elastic axis, from the trailing edg^ to the intersection of the normal with the tip are used in the equation.

The manner of dividing the wing into strips, which has given good results and has been adopted as a standard technique, is also shown in Figure 10. The ith stations are defined as lying at percent span stations .05, .15, .25, .35, .45, .55, .65, .75, .85, .945, and .99. The kth stations are at distances from the root equal to .025, .1, .2, .3, .4, .5, .6, .7, .8, .9, and .99 times the span. This strip definition has also been adopted as the standard for straight wings as well as sweptback planforms. When this division of the planform is used, the matrix defined by $[I_n]$ $[\Delta X_k]$ may be written as

$$[I_n]\,[\Delta X_k] = l \begin{bmatrix} .05 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ .05 & .1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ .05 & .1 & .1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ .05 & .1 & .1 & .1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ .05 & .1 & .1 & .1 & .1 & 0 & 0 & 0 & 0 & 0 & 0 \\ .05 & .1 & .1 & .1 & .1 & .1 & 0 & 0 & 0 & 0 & 0 \\ .05 & .1 & .1 & .1 & .1 & .1 & .1 & 0 & 0 & 0 & 0 \\ .05 & .1 & .1 & .1 & .1 & .1 & .1 & .1 & 0 & 0 & 0 \\ .05 & .1 & .1 & .1 & .1 & .1 & .1 & .1 & .1 & 0 & 0 \\ .05 & .1 & .1 & .1 & .1 & .1 & .1 & .1 & .1 & .095 & 0 \\ .05 & .1 & .1 & .1 & .1 & .1 & .1 & .1 & .1 & .095 & .045 \end{bmatrix}$$

where $l$ = total structural span along the elastic axis.

The length of the elastic axis is assumed to be defined by the distance from the wing-fuselage intersection to the tip, provided the wing-fuselage intersection is not extremely flexible. For wings of moderate sweep, the chordwise location of the elastic axis is not extremely critical, and it has been the practice to assume the 50-percent chord line as the elastic axis, if the true location is not known. For more sharply swept wings, the chordwise location is slightly more critical, since chordwise changes in elastic axis location can cause a more appreciable change in length of the elastic axis. However, it has been found that the location of the elastic axis can usually be estimated accurately enough with only the simplest of calculations.




Figue 10. Geometry for swept wing.

$[I_n]^T [C_i^2][\Delta X_i]$ is equivalent to

$$\frac{L}{10}\begin{bmatrix}
C^2_{.05L} & C^2_{.15L} & C^2_{.25L} & C^2_{.35L} & C^2_{.45L} & C^2_{.55L} & C^2_{.65L} & C^2_{.75L} & C^2_{.85L} & .8C^2_{.945L} & .2C^2_{.99L} \\
0 & C^2_{.15L} & C^2_{.25L} & C^2_{.35L} & C^2_{.45L} & C^2_{.55L} & C^2_{.65L} & C^2_{.75L} & C^2_{.85L} & .8C^2_{.945L} & .2C^2_{.99L} \\
0 & 0 & C^2_{.25L} & C^2_{.35L} & C^2_{.45L} & C^2_{.55L} & C^2_{.65L} & C^2_{.75L} & C^2_{.85L} & .8C^2_{.945L} & .2C^2_{.99L} \\
0 & 0 & 0 & C^2_{.35L} & C^2_{.45L} & C^2_{.55L} & C^2_{.65L} & C^2_{.75L} & C^2_{.85L} & .8C^2_{.945L} & .2C^2_{.99L} \\
0 & 0 & 0 & 0 & C^2_{.45L} & C^2_{.55L} & C^2_{.65L} & C^2_{.75L} & C^2_{.85L} & .8C^2_{.945L} & .2C^2_{.99L} \\
0 & 0 & 0 & 0 & 0 & C^2_{.55L} & C^2_{.65L} & C^2_{.75L} & C^2_{.85L} & .8C^2_{.945L} & .2C^2_{.99L} \\
0 & 0 & 0 & 0 & 0 & 0 & C^2_{.65L} & C^2_{.75L} & C^2_{.85L} & .8C^2_{.945L} & .2C^2_{.99L} \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & C^2_{.75L} & C^2_{.85L} & .8C^2_{.945L} & .2C^2_{.99L} \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & C^2_{.85L} & .8C^2_{.945L} & .2C^2_{.99L} \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & .8C^2_{.945L} & .2C^2_{.99L} \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & .2C^2_{.99L}
\end{bmatrix}$$

Low aspect ratio, highly swept planforms are common for empennage surfaces and present a special case in the treatment of the computed GJ in the area of the root of the surface. This case is different because the actual cross section of the torque box becomes narrower in the area inboard of the elastic axis station where the trailing edge of the torque box intersects the root rib, or fuselage, side. In this area, the trailing edge of the torque box

section which is normal to the elastic axis is actually formed by the root rib. This is a triangular area, shown as a shaded area in Figure 10. The most rigorous application of the method would require that the true values for the ratio of torque box area to perimeter be used in the equations. However, it has been found that an approximation can be made which simplifies the input data and still allows sufficient accuracy to be obtained. This approximate procedure consists of merely assuming that the ratio of torque box area to perimeter continues to follow the same variation in this root area as it follows over the outer span. The GJ is computed under this assumption; then, the GJ computed is not used for design of the sections inboard of the elastic axis station at which the rear spar intersects the fuselage or root rib. Instead, the required GJ is assumed to be that which results from a constant skin gage over the triangular area under consideration. This gage is required to be the same as that determined by the required GJ at the elastic axis station which forms the outboard edge of the triangular area. This results in decreased stiffness requirements in the inboard area, as compared to the actual value. However, experience has shown that the use of the full chord of the surface for the inboard area results in an adequate level of stiffness over the remainder of the span to compensate for the loss in stiffness in the inboard area.

## EXTENSION TO TRANSONIC AND SUPERSONIC SPEEDS

The previous technique is based on incompressible aerodynamic theory and, therefore, is not directly applicable for transonic and supersonic design. The technique used for extending the method to higher speeds is based on observed characteristics of plots of the dimensionless flutter parameter $b\omega_\alpha \sqrt{\mu}/a$ versus mach number. Plots of this parameter for several different planforms are shown in Figures 11 through 13. The area below such a curve represents the region of instability, while the area above the curve is the stable or no-flutter region. It is usually found that aspect ratio, taper ratio, and sweep angle are the primary factors which affect the level and shape of these curves, except in cases where frequency ratios are close to unity, or mass ratios are very low. Therefore, for design purposes, it is assumed that wings having the same aspect ratio, taper ratio, and sweep angle will have approximately the same flutter boundary or plot of $b\omega_\alpha \sqrt{\mu}/a$ vesus mach number.

Straight lines through the origin of such a plot are lines of constant dynamic pressure, or q, for a given wing, and the slope of the line is inversely proportional to the square root of q. Horizontal lines are lines of constant altitude or lines of constant density, with speed of sound times square root of density inversely proportional to $b\omega_\alpha \sqrt{\mu}/a$. As shown by Figures 11 through 13, the curves are always straight lines for mach numbers below the speed range at which compressibility effects begin. This is in line with the known fact that flutter usually occurs at a constant dynamic pressure in the subsonic speed range.

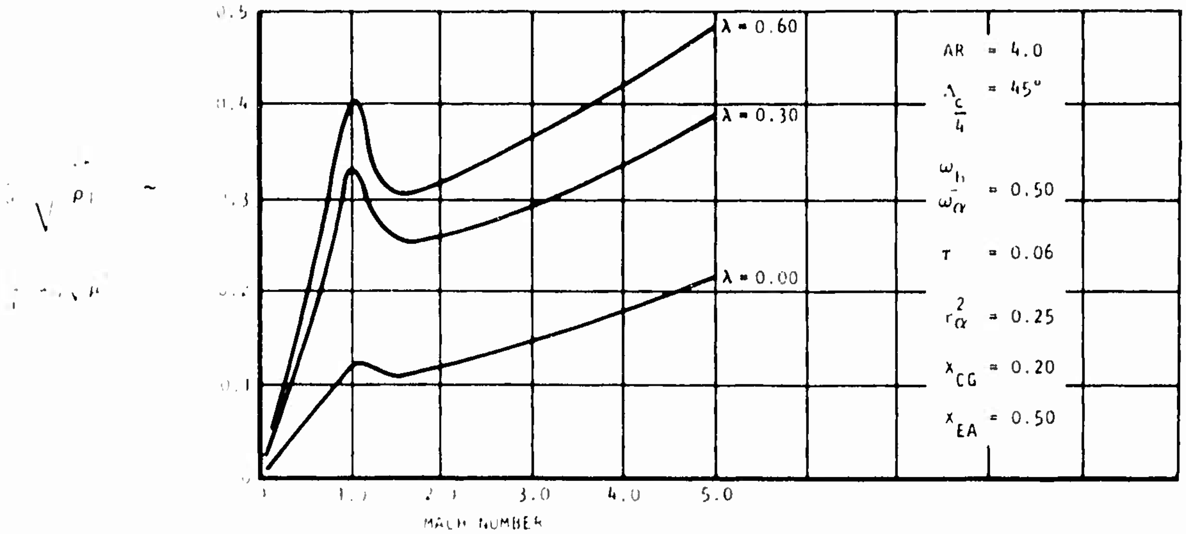



Figure 11. Effect of taper ratio, λ .

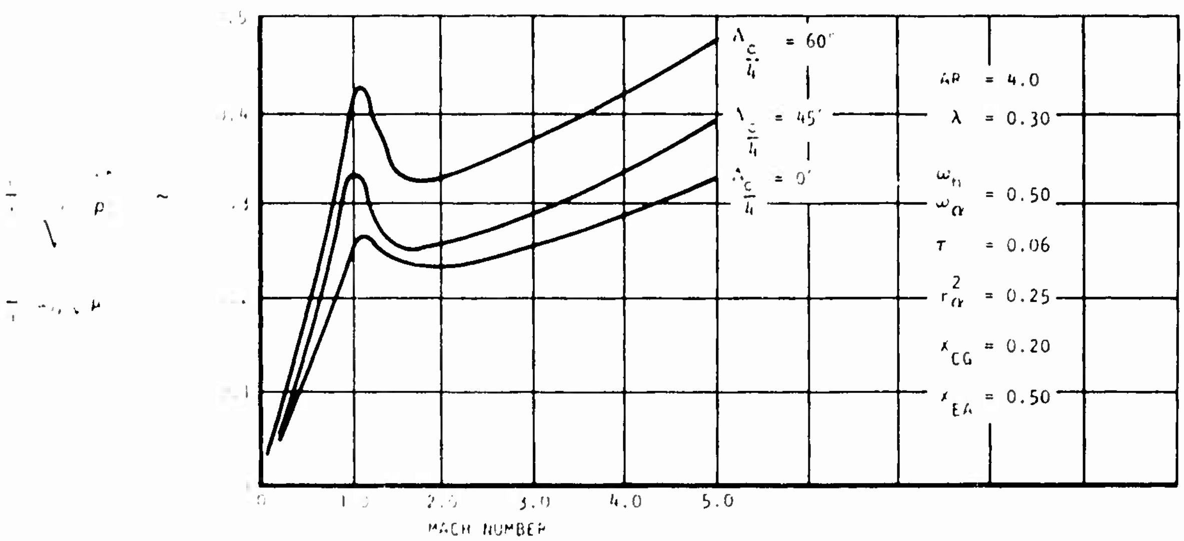



Figure 12. Effect of quarter-chord sweep angle, $\Lambda_{\frac{c}{4}}$ .

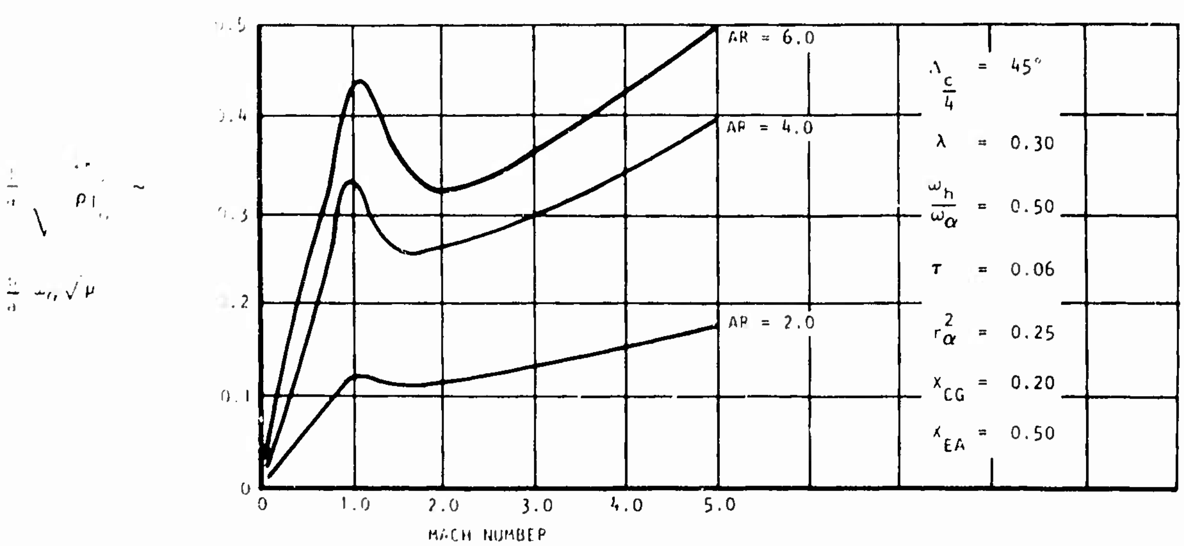



Figure 13. Effect of aspect ratio, AR.

It will now be shown how these characteristics of plots of the dimensionless flutter parameter suggest a technique whereby such plots were used to modify the stiffness criteria developed in the preceding paragraphs. The required stiffness computed by the aforementioned method is directly proportional to dynamic pressure. Therefore, the modification will be effected by computing the proper dynamic pressure to be inserted in the preceding equations so that the stiffness computed will insure flutter stability at the required speeds. This dynamic pressure is determined from the plot of $b\omega_\alpha \sqrt{\mu}/a$ versus mach number for the planform, and from the required speed-altitude curve by the following procedure: An altitude or density scale to be superimposed on the plot of $b\omega_\alpha \sqrt{\mu}/a$ versus mach number may be chosen so that when the required speed-altitude curve is plotted, no points on the flutter boundary lie above the speed-altitude curve, but so that the flutter boundary is tangent to the speed-altitude curve at one or more points. This, of course, assumes that the required speed-altitude curve or speed-density curve is defined by speeds sufficiently above the maximum operating speed of the air vehicle to provide the required margin of safety in flutter stability. The validity of arbitrarily placing the altitude, or density scale, on the flutter boundary plot stems from the fact that $b\omega_\alpha \sqrt{\mu}/a$ is nondimensional. Locating the altitude scale for a given wing merely defines the relative level of $\omega$ which the wing must have in order for the values of $b\omega_\alpha \sqrt{\mu}/a$ to be compatible with the altitude scale. The frequency, $\omega_\alpha$, is directly proportional to the square root of the wing stiffness level so that the choice of the altitude scale may be thought of as defining the required relative stiffness level of the wing.

The altitude scale chosen then defines the dynamic pressure which corresponds to the constant q-line of the flutter bondary. This dynamic pressure is the value to be used in the calculation of the required stiffness for the wing.

## ILLUSTRATIVE EXAMPLE

An example of the procedure to determine the proper dynamic pressure to be used in equation 23 illustrates the technique. This approach has been programmed in subroutine QSUB in the flutter and temperature module.

It will be assumed that it is desired to compute the stiffness requirements for a surface with aspect ratio = 4.0, taper ratio = 0.3, and a sweepback of the quarter chord line of 45 degrees. The flight operational requirements will be assumed to be those described by the altitude mach number curve in Figure 14.

It will be further assumed that a 20-percent margin in speed and a 44-percent margin in dynamic pressure must be added to the operational requirement to obtain the flutter design requirement. The general flutter boundary for the planform is taken from Figure 15. The requirement for 20-percent margin in speed determines that a sea-level altitude line must intersect the flutter boundary at M = 0.96.

This is shown in Figure 15 as line AB. A line drawn through the origin and the point of intersection of the sea-level line and the flutter boundary represents the required flutter design q of 1,370 psf. This is shown as line BD in Figure 15. It may be seen that all points on the flutter boundary for mach numbers greater than 0.96 lie below their required design q-line. However, at mach numbers below 0.96, a portion of the boundary lies above the required q-line. Hence, this relative location of the design q-line on the plot will not result in satisfaction of the requirement for margin of safety in dynamic pressure. Therefore, a new line must be drawn on the plot to represent the design q-line. This is line CE in Figure 15 and is drawn so that it is just tangent to the flutter boundary. This line now represents the design dynamic pressure, q = 1,370 psf. The corresponding new sea-level line may be defined by drawing a horizontal line, FG in Figure 15, so that it intersects the design q-line at M = 0.96. The proper relation between the altitude scale and the general flutter boundary has now been established, and lines IK and KE represent the design requirements obtained by adding the necessary margin of safety in speed and dynamic pressure to the operational curve. The final step is to determine what value of dynamic pressure is represented by the straight-line portion of the flutter boundary, designated by line CH in Figure 15. Since the slopes of straight lines through the origin are inversely proportional to the square root of dynamic pressure, the dynamic pressure which corresponds to line CH is given by the dynamic pressure of line CG multiplied by the square of the ratio of the slope of line CG to the slope of line CH. For the subject example, this ratio of the slopes is 0.357/0.307 = 1.16, and the square of the ratio is 1.35. Therefore, the dynamic pressure corresponding to line CH is 1.35 x 1,370 = 1,850 psf. If this value of dynamic pressure is used in equation 36, the resulting stiffness will be the optimum distribution which will provide the required margin of safety for the operational curve of Figure 14. For limitations inherent in the programmed selection of these critical dynamic pressure, refer to note on page 141.

This example shows that compressibility effects can greatly increase the stiffness required to prevent flutter. In this example, the method indicates that a 35-percent increase in stiffness is necessitated by compressibility effects. Clearly, when increments of this magnitude are involved, optimum design cannot be achieved if the unmodified stiffness criterion for the incompressible speed range is used.

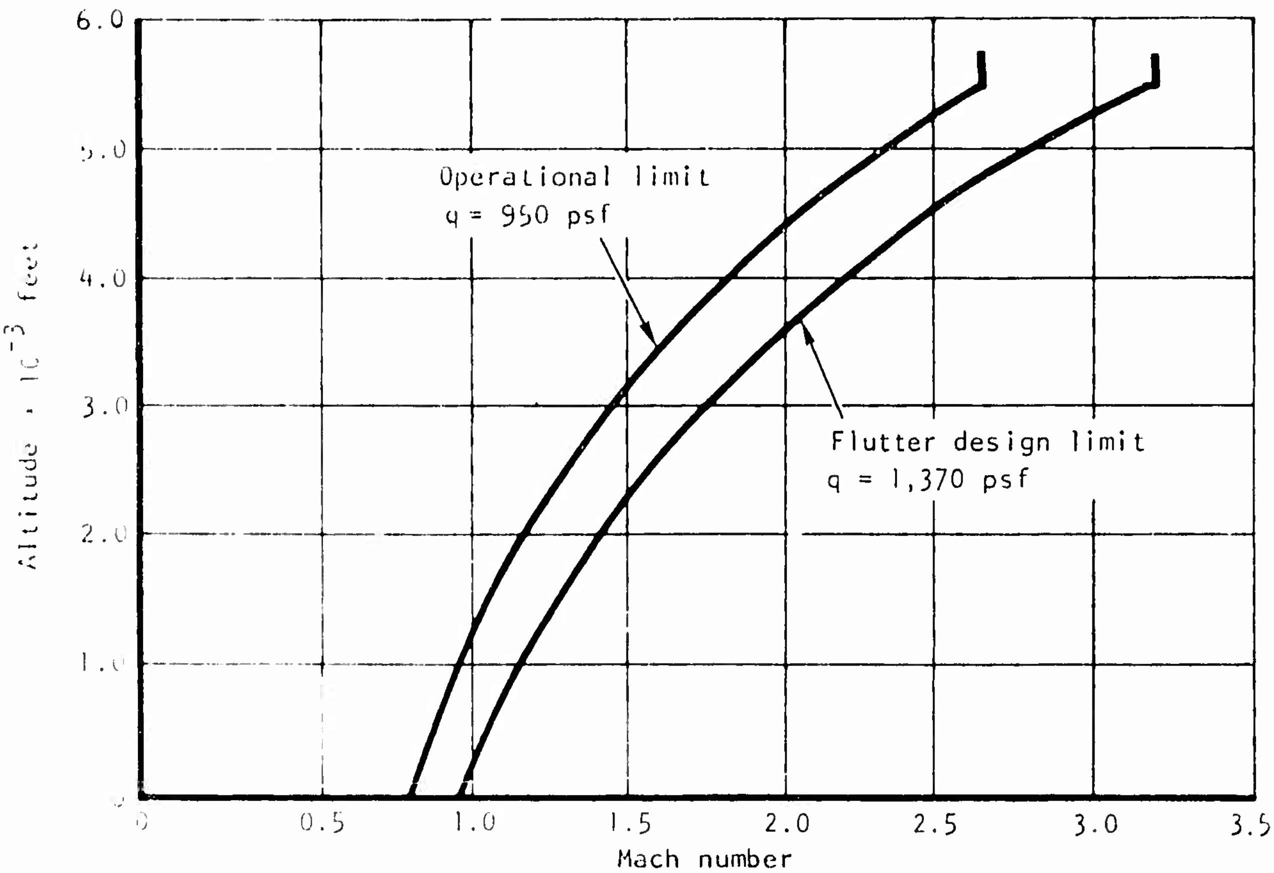



Figure 14. Altitude and mach number curves for example No. 1.

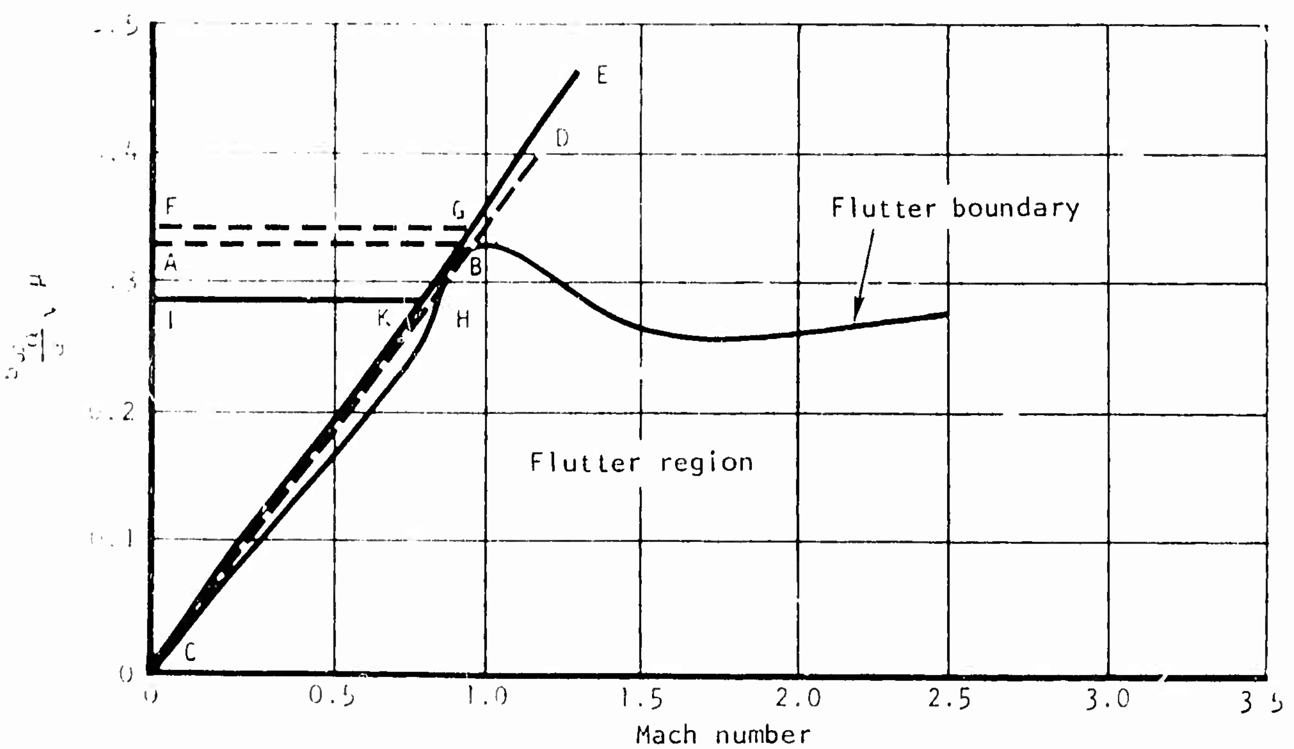



Figure 15. Flutter boundary and design requirements for example No. 1.

## INTERPOLATION FOR FLUTTER BOUNDARIES

It is often necessary to compute stiffness requirements for planforms having combinations of aspect ratio, taper ratio, and sweep angle which are not identical to any of the combinations represented by Figures 11 through 13. However, these parameters for most practical surfaces do fall within the ranges of aspect ratio, taper ratio, and sweep which correspond to the figures. For these cases, the flutter boundary for the planforms were approximated by interpolation between the curves in Figures 11 through 13. Tabulated values of the flutter parameter versus mach number are shown in Table 7.

To illustrate the interpolation technique, the example planform has an aspect ratio of 5.0, a taper ratio of 0.25, and a sweep angle of 30 degrees. The value for the altitude stiffness parameter at a particular mach number is given by the figurative formula:

$$\begin{pmatrix} AR = 5 \\ \lambda = .25 \\ \Lambda_{c/4} = 30^\circ \end{pmatrix} = \frac{\begin{pmatrix} AR = 5 \\ \lambda = .3 \\ \Lambda_{c/4} = 45^\circ \end{pmatrix}\begin{pmatrix} AR = 4 \\ \lambda = .25 \\ \Lambda_{c/4} = 45^\circ \end{pmatrix}\begin{pmatrix} AR = 4 \\ \lambda = .3 \\ \Lambda_{c/4} = 30^\circ \end{pmatrix}}{\begin{pmatrix} AR = 4 \\ \lambda = .3 \\ \Lambda_{c/4} = 45^\circ \end{pmatrix}}$$

This procedure is used in subroutine SVFTAB to obtain the flutter parameter curve for the given wing, horizontal tail, and vertical tail planform geometries.

## T-TAIL FLUTTER METHODOLOGY

Although some interaction in stiffness requirements of the horizontal and vertical tail surfaces may occur, the stiffness level of the vertical tail is usually the most important for T-tail flutter, provided, of course, that the horizontal tail has adequate stiffness to prevent flutter when it is treated as an independent surface. It is therefore assumed that the preliminary stiffness requirements for the horizontal tail are adequately determined by the method described for the prevention of surface flutter.

TABLE 7. DATA FOR FLUTTER PARAMETER VERSUS MACH NUMBER CURVES

| Curve | 1 | | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|---|
| AR<br>$\lambda$<br>$\Lambda_{c/4}$ | 4.0<br>0.3<br>45° | | 2.0<br>0.3<br>45° | 6.0<br>0.3<br>45° | 4.0<br>0.3<br>0° | 4.0<br>0.3<br>60° | 4.0<br>0.0<br>45° | 4.0<br>0.6<br>45° |
| Point | Mach No. | TBP | TAR2 | TAR6 | TSB0 | TSB60 | TTR0 | TTR60 |
| 1 | 0.200 | 0.060 | 0.024 | 0.082 | 0.052 | 0.073 | 0.025 | 0.077 |
| 2 | 0.400 | 0.120 | 0.048 | 0.164 | 0.104 | 0.146 | 0.050 | 0.154 |
| 3 | 0.600 | 0.180 | 0.072 | 0.146 | 0.156 | 0.219 | 0.075 | 0.231 |
| 4 | 0.800 | 0.240 | 0.096 | 0.328 | 0.208 | 0.292 | 0.100 | 0.308 |
| 5 | 0.825 | 0.255 | 0.099 | 0.342 | 0.214 | 0.307 | 0.103 | 0.322 |
| 6 | 0.850 | 0.275 | 0.102 | 0.360 | 0.220 | 0.325 | 0.106 | 0.340 |
| 7 | 0.875 | 0.295 | 0.105 | 0.385 | 0.226 | 0.343 | 0.109 | 0.360 |
| 8 | 0.900 | 0.310 | 0.108 | 0.400 | 0.232 | 0.360 | 0.112 | 0.375 |
| 9 | 0.925 | 0.322 | 0.111 | 0.410 | 0.238 | 0.376 | 0.115 | 0.387 |
| 10 | 0.950 | 0.327 | 0.113 | 0.418 | 0.244 | 0.390 | 0.118 | 0.395 |
| 11 | 0.975 | 0.330 | 0.115 | 0.425 | 0.250 | 0.401 | 0.121 | 0.400 |
| 12 | 1.000 | 0.332 | 0.117 | 0.430 | 0.255 | 0.410 | 0.123 | 0.403 |
| 13 | 1.025 | 0.333 | 0.119 | 0.433 | 0.259 | 0.417 | 0.125 | 0.404 |
| 14 | 1.050 | 0.332 | 0.120 | 0.435 | 0.263 | 0.422 | 0.126 | 0.405 |
| 15 | 1.075 | 0.329 | 0.121 | 0.437 | 0.266 | 0.424 | 0.127 | 0.404 |
| 16 | 1.100 | 0.325 | 0.122 | 0.436 | 0.268 | 0.425 | 0.128 | 0.402 |
| 17 | 1.125 | 0.320 | 0.123 | 0.437 | 0.269 | 0.423 | 0.127 | 0.395 |
| 18 | 1.150 | 0.314 | 0.122 | 0.435 | 0.268 | 0.420 | 0.126 | 0.385 |
| 19 | 1.175 | 0.307 | 0.121 | 0.433 | 0.266 | 0.416 | 0.125 | 0.370 |
| 20 | 1.200 | 0.300 | 0.120 | 0.430 | 0.263 | 0.410 | 0.124 | 0.358 |
| 21 | 1.300 | 0.278 | 0.117 | 0.413 | 0.252 | 0.382 | 0.115 | 0.330 |
| 22 | 1.400 | 0.267 | 0.114 | 0.390 | 0.245 | 0.354 | 0.111 | 0.310 |
| 23 | 1.500 | 0.262 | 0.112 | 0.365 | 0.240 | 0.337 | 0.110 | 0.306 |
| 24 | 1.600 | 0.259 | 0.111 | 0.350 | 0.237 | 0.330 | 0.111 | 0.305 |
| 25 | 1.700 | 0.257 | 0.110 | 0.337 | 0.235 | 0.327 | 0.112 | 0.307 |
| 26 | 1.800 | 0.257 | 0.110 | 0.330 | 0.234 | 0.325 | 0.114 | 0.310 |
| 27 | 1.900 | 0.258 | 0.111 | 0.323 | 0.235 | 0.327 | 0.116 | 0.313 |
| 28 | 2.000 | 0.260 | 0.112 | 0.320 | 0.236 | 0.330 | 0.118 | 0.316 |
| 29 | 2.100 | 0.262 | 0.114 | 0.320 | 0.237 | 0.333 | 0.121 | 0.320 |
| 30 | 2.200 | 0.264 | 0.116 | 0.322 | 0.238 | 0.337 | 0.124 | 0.324 |
| 31 | 2.300 | 0.266 | 0.118 | 0.325 | 0.239 | 0.341 | 0.127 | 0.328 |
| 32 | 2.400 | 0.269 | 0.120 | 0.328 | 0.241 | 0.345 | 0.130 | 0.333 |
| 33 | 2.500 | 0.272 | 0.122 | 0.333 | 0.243 | 0.349 | 0.133 | 0.338 |
| 34 | 3.000 | 0.292 | 0.130 | 0.360 | 0.256 | 0.372 | 0.148 | 0.364 |
| 35 | 3.500 | 0.315 | 0.139 | 0.392 | 0.273 | 0.397 | 0.164 | 0.392 |
| 36 | 4.000 | 0.339 | 0.149 | 0.425 | 0.291 | 0.423 | 0.181 | 0.422 |
| 37 | 4.500 | 0.364 | 0.160 | 0.460 | 0.310 | 0.451 | 0.199 | 0.454 |
| 38 | 5.000 | 0.390 | 0.172 | 0.498 | 0.330 | 0.480 | 0.217 | 0.486 |

The important problem remaining then is to develop a method for determining preliminary stiffness requirements for the vertical tail. Experimental flutter data from wind tunnel flutter models of a typical T-tail are given in Reference 19. These data are used as the basis of a stiffness criteria for the vertical tail in T-tail configurations. In order to develop the criteria, dynamic similarity relations were used to allow scaling the flutter model results to the desired full-scale tail. This establishes the required stiffness level. The optimum distribution of the stiffness is then determined by methods described for the prevention of surface flutter.

## SCALING FROM TEST RESULTS

In order to use the results of Reference 19 for different size horizontal tails relative to the vertical tail, and different aspect ratio vertical tails, some assumptions must be made. Scaling the entire model uniformly is considered as included in the general scaling relations developed. The assumptions are:

1. Flutter of the full scale occurs at the same reduced velocity as the model, or

$$\left(\frac{V}{b\omega}\right)_{MOD} = \left(\frac{V}{b\omega}\right)_{FS} \tag{37}$$

$$\omega_{FS} = \omega_{MOD} \frac{b_{MOD}}{b_{FS}} \frac{V_{FS}}{V_{MOD}} \tag{38}$$

where

$V$ = flutter velocity

$b$ = reference length, average chord

$\omega$ = flutter frequency

subscripts

FS = refers to full scale

MOD = refers to model

2. Assume mass ratio effects are negligible, so it can be assumed that

$$\left(\frac{V_{FS}}{V_{MOD}}\right)^2 = \frac{q_{FS}}{q_{MOD}} \tag{39}$$

where

$q$ = flutter dynamic pressure

3. Assume that scaling the cantilevered torsion frequency of the vertical tail, with horizontal tail attached, gives the proper flutter frequency

4. The cantilevered torsion frequency of the vertical tail is determined entirely by the GJ of the vertical tail and the yawing moment of inertia of the horizontal tail

From assumptions 1 and 2,

$$\omega_{FS}^2 = \omega_{MOD}^2 \left(\frac{b_{MOD}}{b_{FS}}\right)^2 \left(\frac{q_{FS}}{q_{MOD}}\right) \tag{40}$$

In order to determine the optimum GJ distribution to provide the required frequency, assume an inertia torque at the tip, T, of the vertical tail elastic axis:

$$T = \omega^2 I_\psi \theta_T \tag{41}$$

where

$I_\psi$ = inertia of the horizontal tail

$\theta_T$ = change in twist angle

Also, assume that an optimum GJ will result if skin thickness is distributed such that the torsion stress is constant over the span of the vertical tail.

$$\{T_K\} = 2 \left[A_K\right] \left[\sigma_K\right] \{t_K\} \tag{42}$$

$$\{t_K\} = \frac{1}{2} \left[\frac{1}{A_K}\right] \left[\frac{1}{\sigma_K}\right] \{t_K\} \tag{43}$$

where

$t_K$ = skin thickness

If $T_K$ is only due to torque at the tip, then $T_K$ is a constant over the span such that

$$\{t_K\} = \frac{T}{2\sigma} \left\{\frac{1}{A_K}\right\} \tag{44}$$

from the definition of GJ,

$$\{J\} = 4 \begin{bmatrix} \frac{A_K^2}{S_K} \end{bmatrix} \{t_K\} \tag{45}$$

Equation 46 is then obtained by inserting equation 44 in equation 25

$$\{J\} = \frac{2T}{\sigma} \left\{\frac{A_K}{S_K}\right\} \tag{46}$$

From equation 35,

$$\theta_T = \frac{\sigma}{2G} \lfloor \Delta X_K \rfloor \left\{\frac{S_K}{A_K}\right\} \tag{47}$$

From equations 41 and 47,

$$T = \omega^2 I_\psi \frac{\sigma}{2G} \lfloor \Delta X_K \rfloor \left\{\frac{S_K}{A_K}\right\} \tag{48}$$

Then, from equations 46 and 48,

$$G\{J\} = \omega^2 I_\psi \left( \lfloor \Delta X_K \rfloor \left\{\frac{S_K}{A_K}\right\} \right) \left\{\frac{A_K}{S_K}\right\} \tag{49}$$

From equations 40 and 49,

$$\{GJ\} = \omega_{MOD}^2 \left(\frac{b_{MOD}}{b_{FS}}\right)^2 \left(\frac{q_{FS}}{q_{MOD}}\right) I_\psi \left( \lfloor \Delta X_K \rfloor \left\{\frac{S_K}{A_K}\right\} \right) \left\{\frac{A_K}{S_K}\right\} \tag{50}$$

In order to include some of the effects of the change in aerodynamic forces on the vertical tail due to change in vertical tail aspect ratio, $e_{eT}$, the modified stiffness parameter can be used. The right-hand side of equation 50 would be multiplied by $e_{eT}/e_{eTMOD}$ to introduce the effects of vertical planform. The planform geometry parameter, $e_{eT}$, is defined by equation 51.

$$e_{eT} = \frac{25.88}{(1+0.8/AR)^2}\left[0.4 + 0.7 \cos (\Lambda c/4 - 10°)\right] \tag{51}$$

Model parameters are combined into a single constant $C_{TT}$.

$$C_{TT} = \frac{(\omega_{MOD}\ b_{MOD})^2}{q_{MOD}\ e_{eT_{MOD}}} \tag{52}$$

The solution for torsional stiffness can then be written in terms of the constant, $C_{TT}$, and the full-scale parameters. Equation 53 is the form used in subroutine GJTT of the wing and empennage weight estimation module.

$$\{GJ\} = \frac{C_{TT}\ e_{eT}\ q\ I_\psi}{b^2}\left(\lfloor \Delta X_K \rfloor \left\{\frac{S_K}{A_K}\right\}\right)\left\{\frac{A_K}{S_K}\right\} \tag{53}$$

where

$GJ$ = torsional stiffness, lb-in.$^2$

$C_{TT}$ = model scaling parameter, in.$^6$/lb$^2$/sec$^2$

$e_{eT}$ = planform geometry parameter, psi

$q$ = dynamic pressure at flutter, lb/in.$^2$, this is the flutter design pressure which includes safety factor on flutter speed

$I_\psi$ = horizontal tail yaw inertia, lb-in.-sec$^2$

$b$ = average chord of vertical tail, in.

The flutter model parameters in equation 52 are obtained from Reference 19. Data from model numbers 1 through 7 were used for the case for 15-degree horizontal tail dihedral. Data from model numbers 8 and 9 were used for 0-degree dihedral. The values of $C_{TT}$ versus mach number are shown in Figure 16.

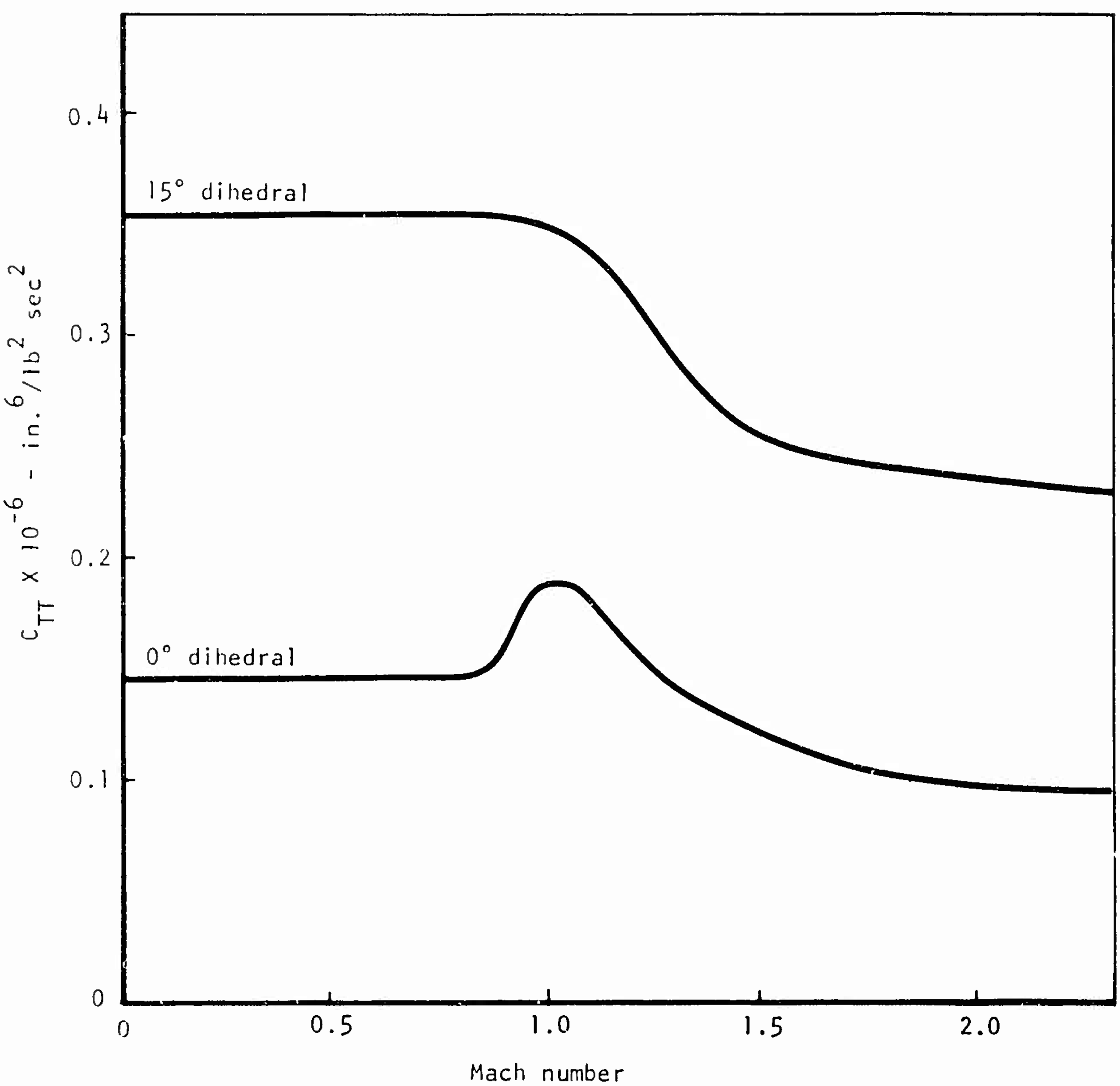



Figure 16. Model scaling parameter, $C_{TT}$, versus mach number for T-tails.

No data are given in Reference 19 for mach numbers above 1.3 or below 0.7. The straight line extrapolation below M = 0.7 is considered reliable, since subsonic flutter occurs at approximately a constant dynamic pressure for most surfaces. The extrapolation above M = 1.3, however, is only based on intuition and observation of supersonic flutter trends for other types of surfaces. This is not expected to be a significant problem, since critical flutter conditions almost always occur at high subsonic or transonic mach numbers.

In using the values of $C_{TT}$ for preliminary design, it must be noted that the minimum allowable torsion stiffness level will be determined by the largest product of $C_{TT}$ and q for cases where q varies with mach numbers.

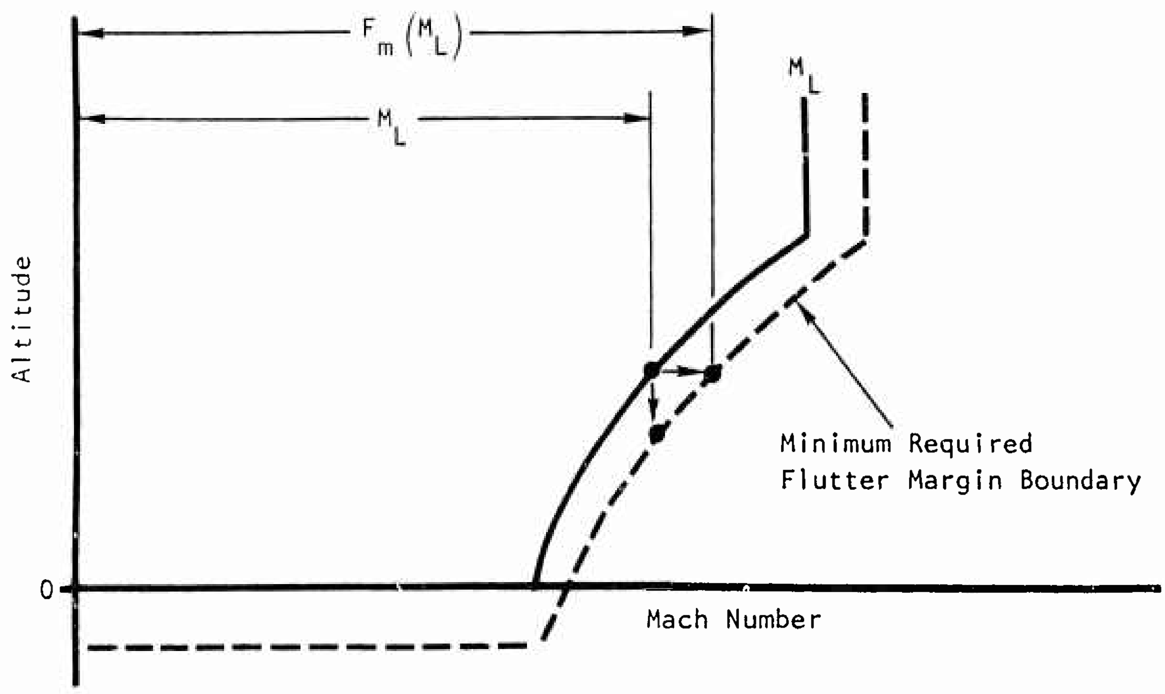
FLUTTER MARGIN

To insure safety, there are two distinct definitons of flutter margin for evaluating vehicle limit speed-altitude profile. The flutter margin, $F_m$, for a given profile point is defined as follows:

1. $M_{Design} = (F_m)\ (M_L)$, constant altitude

2. $q_{Design} = (F_m)^2\ (q_L)$, constant mach number

A graphical significance of this definition is shown in the following sketch.




For constant dynamic pressure profiles, there is no difference between these two definitions for points above sea level. For increasing dynamic pressure profiles and along the sea level line, the second definition establishes the flutter margin boundary.

## SYMBOLS USED IN FLUTTER METHODOLOGY EQUATIONS

| | |
|---|---|
| $A$ | Torque box cross-sectional area, in. |
| $AR$ | Aspect ratio |
| $a$ | Speed of sound, ft/sec |
| $b$ | Reference chord, in. |
| $C$ | Chord, in. |
| $C_{\ell\alpha}$ | Lift curve slope |
| $C_{TT}$ | Model parameter, in.$^6$/lb$^2$/sec$^2$ |
| $e$ | Chordwise eccentricity - measure of distance between elastic axis and chenter of pressure |
| $e_e$ | Flutter-stiffness index, lb/in.$^2$ (refer to equation 22) |
| $e_{eT}$ | Flutter-stiffness index, lb/in.$^2$ (refer to equation 51) |
| $G$ | Material shear modulus, lb/in.$^2$ |
| $I_\psi$ | Horizontal tail yaw inertia, lb-in.-sec$^2$ |
| $J$ | Torsion constant of torque box, in.$^4$ (refer to equations 32 and 45) |
| $K$ | Constants, lb/in.$^2$ |
| $L$ | Structural length of elastic axis, in. |
| $M$ | Mach number |
| $q$ | Dynamic pressure, lb/in.$^2$ or lb/ft$^2$ |
| $S$ | Perimeter of torque box, in. |
| $T$ | Total torque acting on a section, in.-lb |
| $t$ | Structure material thickness, in. |
| $V_a$ | Velocity of airstream, miles/hr |

$X$ — Spanwise coordinate measured along elastic axis, in.

$\theta$ — Wing twist angle, radians

$\theta'$ — Section increment in twist angle, radians/in.

$\Lambda$ — Sweepback angle, deg

$\lambda$ — Taper ratio

$\mu$ — Mass ratio, ratio of surface weight to weight of air enclosed by a circumscribed cylinder

$\omega$ — Frequency, radians/sec

General Subscripts

FS — Denotes full scale

i — Index of surface section used to define aerodynamic terms

K — Index of surface section used to define structure terms

MOD — Denotes model

Matrix Symbols

$\lfloor \; \rfloor$ — Row matrix

$[ \; ]$ — Square or rectangular matrix

$\lceil \; \rfloor$ — Diagonal matrix

$\{ \; \}$ — Column matrix

$\left[ I_n \right]$ — Integrating matrix made up of unit elements on and below the principal diagonal, and zero elements above the principal diagonal

$\left[ I_n \right]^T$ — Integrating matrix made up of unit elements on and above the principal diagonal, and zero elements below the principal diagonal

# Section III

## PROGRAM DESCRIPTION

### GENERAL DISCUSSION

The main functions of the flutter and temperature program module are:

1. To determine design data at the critical flutter point for the wing, horizontal tail, and vertical tail

2. To determine structure temperature at the vehicle speeds and altitudes where either loads or flutter are evaluated

Specific calculating functions are divided into separate subsets of routines which are called by the module control routine OLAY3.

### THERMAL PROGRAM FUNCTIONS

Seven function subroutines called by subroutine TEMPER are used to calculate equilibrium skin temperatures at the design loading conditions and at the critical flutter speeds.

These routines include the following:

- PRESH Calculates local pressure at a given geopotential altitude
- TEMALT Calculates local temperature for a standard atmosphere at a given geopotential altitude
- TTO Calculates total temperature at a given vehicle speed and altitude
- SOLARG Calculates solar flux at a given geopotential altitude
- TBL Calculates adiabatic wall temperature based on an assumed skin temperature
- HBL Calculates aerodynamic heat transfer coefficient based on assumed adiabatic wall and skin temperatures
- TSKIN Calculates skin temperature by solving the heat equation

## FLUTTER PROGRAM FUNCTIONS

Routines called by subroutine WHVQQ are used to determine the design dynamic pressure, corrected for compressibility effects, and the corresponding material shear modulus at the design point.

The following three routines called by WHVQQ are used for determining temperatures and atmospheric properties:

- PRESH Calculates local pressure at a given geopotential altitude; returns pressure in lb/ft$^2$ to the using statement.

- TEMALT Calculates local temperature for a standard atmosphere at a given geopotential altitude; returns temperature in degrees Rankine to the using statements.
- TEMPER Calculates skin temperature for a given mach number and altitude; returns skin temperature in degrees Fahrenheit as part of the argument list. Arguments, in the order of the calling statement, are shown in the following list; the letter "I" is used to designate input argument, and "C" to denote calculated variable.

| Argument | Type | Description |
|---|---|---|
| XM | I | Vehicle mach number |
| AL | I | Local altitude, ft |
| PRE | C | Local static pressure, lb/in.$^2$ |
| TLOC | C | Local static temperature, $^\circ$ R |
| TTOT | C | Total temperature, $^\circ$ R |
| SFX | C | Solar flux, BTU/hr/ft$^2$ |
| TSKR | C | Equilibrium skin temperature, $^\circ$ R |
| TSF | C | Equilibrium skin temperature, $^\circ$ F |
| IRE | C | Error indicator |

In addition to the three aforementioned routines, WHVQQ uses the following routines to set up materials data and flutter parameters:

- WHVMAT Enters the material library files and sets up tables of compression yield stress and shear modulus versus temperature for the wing, horizontal tail, and vertical tail structural materials.

- SVFTAB Interpolates the flutter parameter tables to obtain flutter parameter versus mach number for each of the surfaces.

- QINC Calculates incompressible dynamic pressure for a given mach number, local pressure, and local temperature.

- QSUB Calculates design dynamic pressure, corrected for compressibility effects, at each of the speed-altitude profile points.

## LOGIC FLOW

The calling-called matrix for the program showing interdependence of routines is shown in Figure 17.

| Calling \ Called | TEMPER | WHVQQ | PRESH | TEMALT | TTO | SOLARG | TBL | HBL | TSKIN | WHVMAT | SVFTAB | QINC | QSUB |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| OLAY 3 | X | X | | | | | | | | | | | |
| TEMPER | | | X | X | X | X | X | X | X | | | | |
| WHVQQ | X | | X | X | | | | | | X | X | X | X |

Figure 17. Calling-called matrix for flutter and temperature module.

Usage of the flutter and temperature routines within the flutter and temperature module is depicted in Figure 18. Logic flow within the temperature program is shown in Figure 19. Logic flow within the flutter program is shown in Figure 20.

Temperature data calculated by the flutter and temperature module are used both by the module itself in establishing materials properties for subsequent flutter evaluations and by other modules in setting up materials properties for structural synthesis.

## MODULE INTERFACES, INPUT, AND OUTPUT

The flow of data to this module and the usage of data calculated by this module are shown in Figure 21.

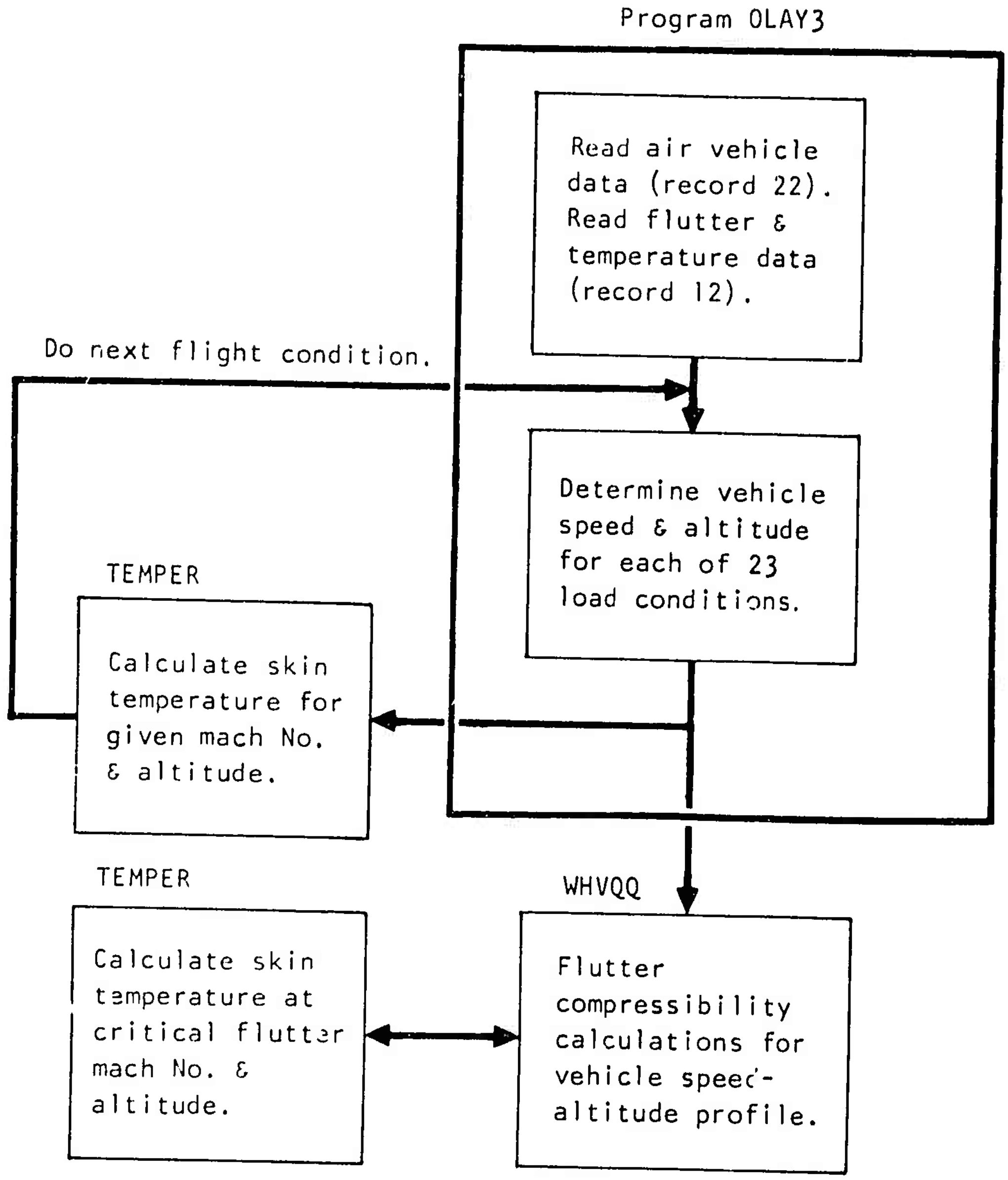



Figure 18. Temperature and flutter routine usage in flutter and temperature module.

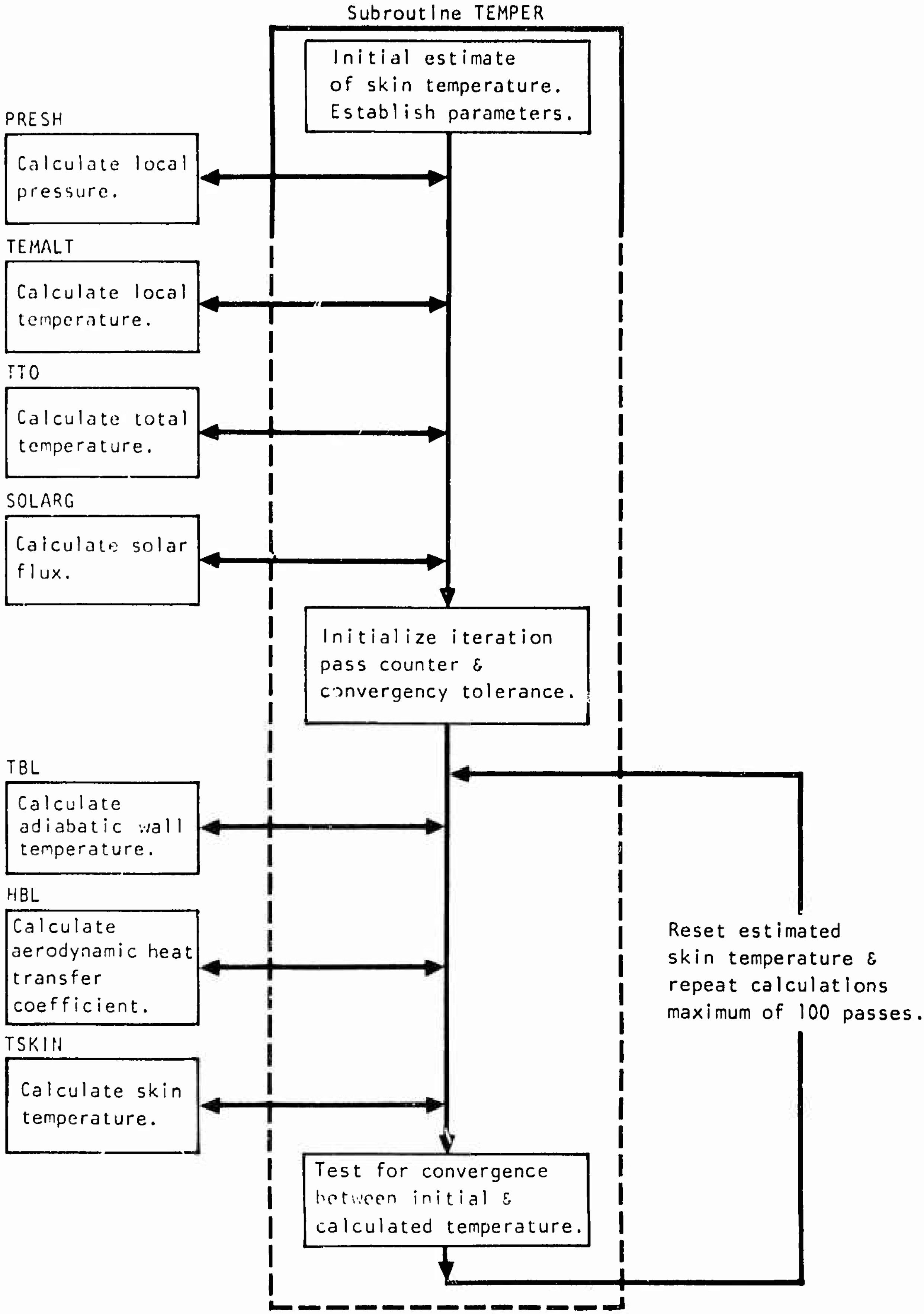



Figure 19. Logic flow diagram for temperature routines.

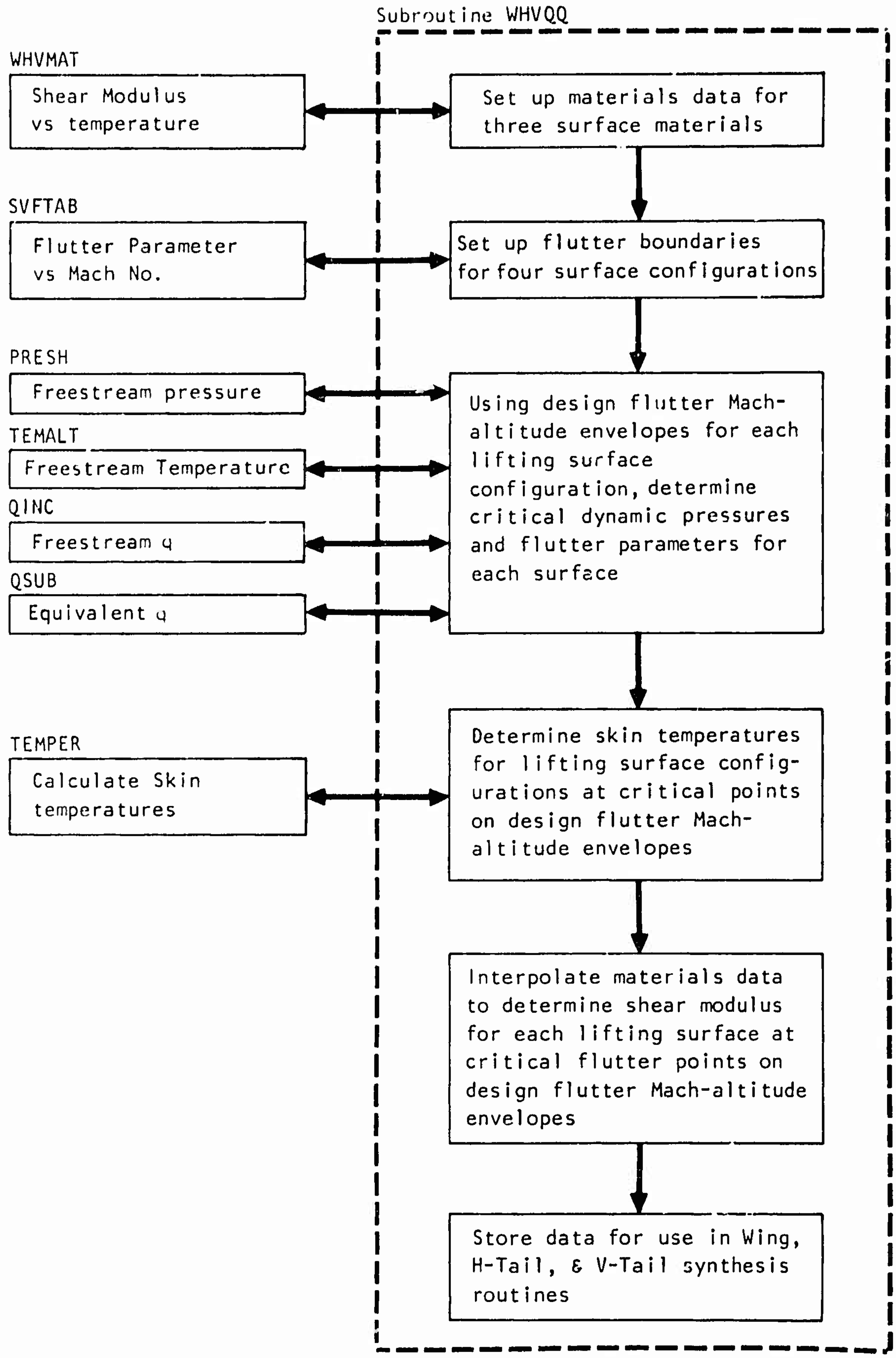



Figure 20. Logic Flow Diagram for Flutter Routines.

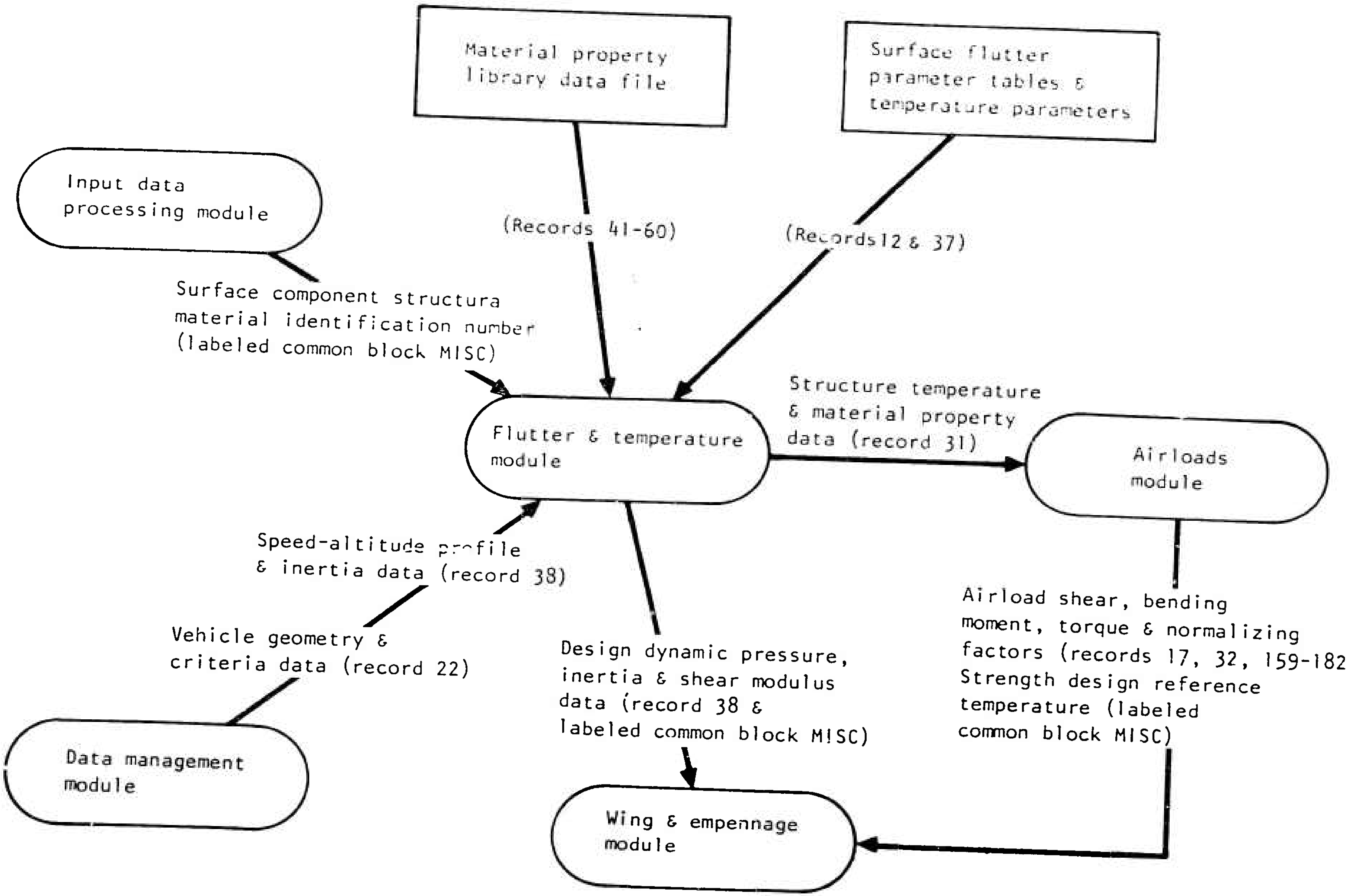



Figure 21. Flow of flutter and temperature module related data.

Figure 22 depicts the flow of flutter-related data to and from the module. Speed altitude profile data input by the user is expanded in the data management module. Figure 23 shows the relationship between input points, loads evaluation points, and flutter evaluation points.

Routines which evaluate flutter considerations are integrated into different modules of SWEEP. A cross-reference list of flutter evaluation and data processing routines within these modules is shown in Table 8.

GENERAL MAPS

Data storage and transmittal are accomplished through the use of common, labeled common, mass storage files, and argument lists. Mass storage records are read into and written from regions in common or from program regions.

Common consists of 762 cell which are divided into the major regions shown in Table 9. Table 10 presents an alphabetical listing of arrays and variables within the common region. Type designates whether variable is input (I) or calculated (C). When the variables in this table are subsets of larger arrays, the higher order array is referenced in brackets. Tables 11 and 12 are expanded explanations of arrays which are not explained in the alphabetical listing.

Labeled common blocks IPRINT and MISC are used to transmit print controls and design data, respectively. IP(41) in the IPRINT block is used to designate output print of design data generated in this module. Variables in the MISC block pertinent to this module are shown in Table 14.

Mass storage file records used in this module are shown in Table 15. The SPAL and TM arrays are read into and written from program regions. The SPAL array is described in Table 17. The TM array is described under the description of subroutine WHVMAT.

Arguments used to transmit data to the temperature calculating routines are shown in the following list in the order of the calling statement. The letter I is used to designate input argument, and C to denote calculated variable.

| Argument | Type | Description |
|---|---|---|
| XMACH | I | Vehicle mach number |
| ALT | I | Local altitude, ft |
| PRESSI | C | Local static pressure, psi |
| TLOC | C | Local static temperature, deg R |
| TTOT | C | Total temperature, deg R |

| Argument | Type | Description |
|---|---|---|
| SUN | C | Solar flux, BTU/hr/ft$^2$ |
| XSKIN | C | Equilibrium skin temperature, deg R |
| XSKINF | C | Equilibrium skin temperature, deg F |
| IER | C | Error indicator |

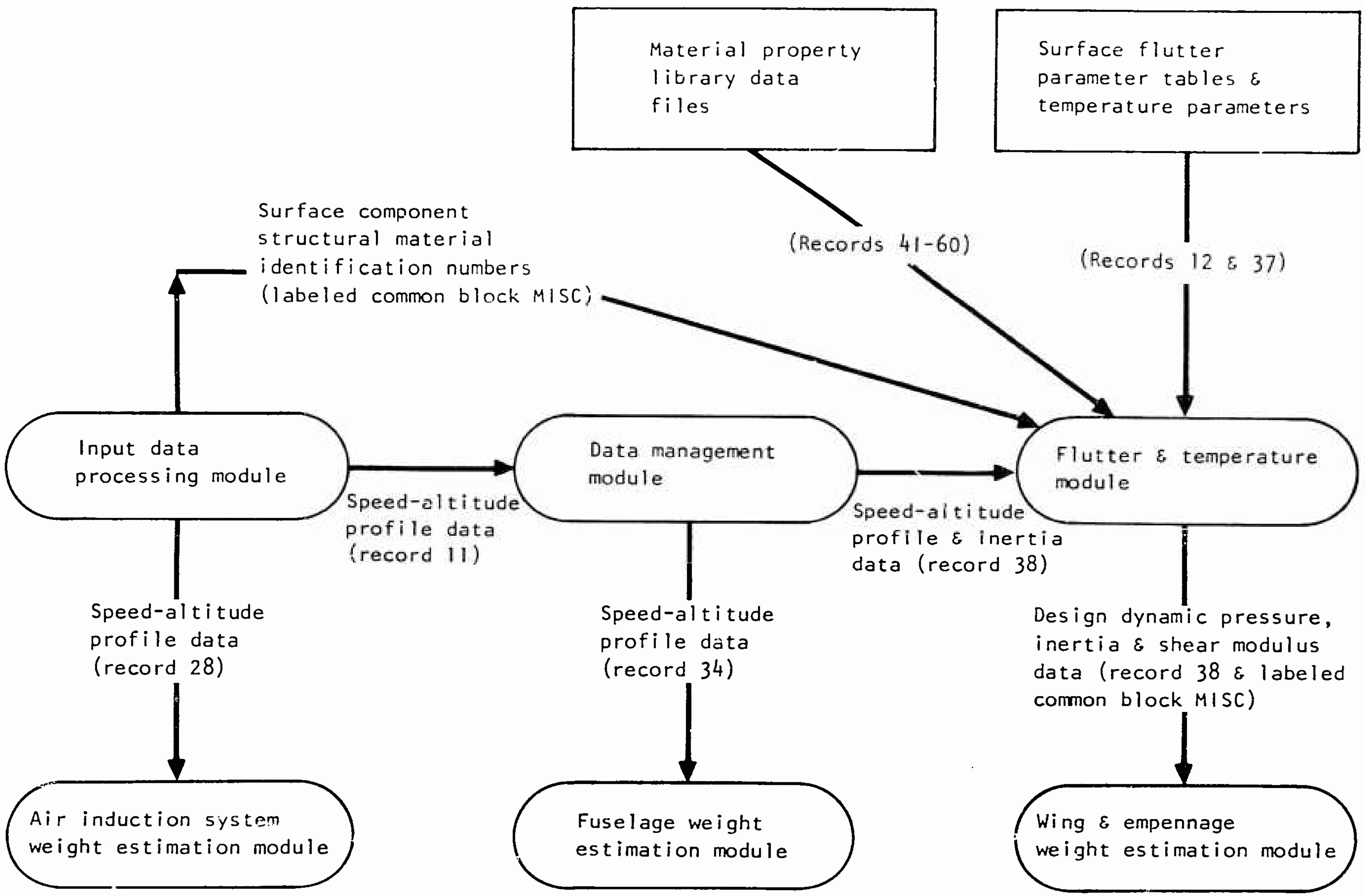



Figure 22. Flow of flutter-related data.

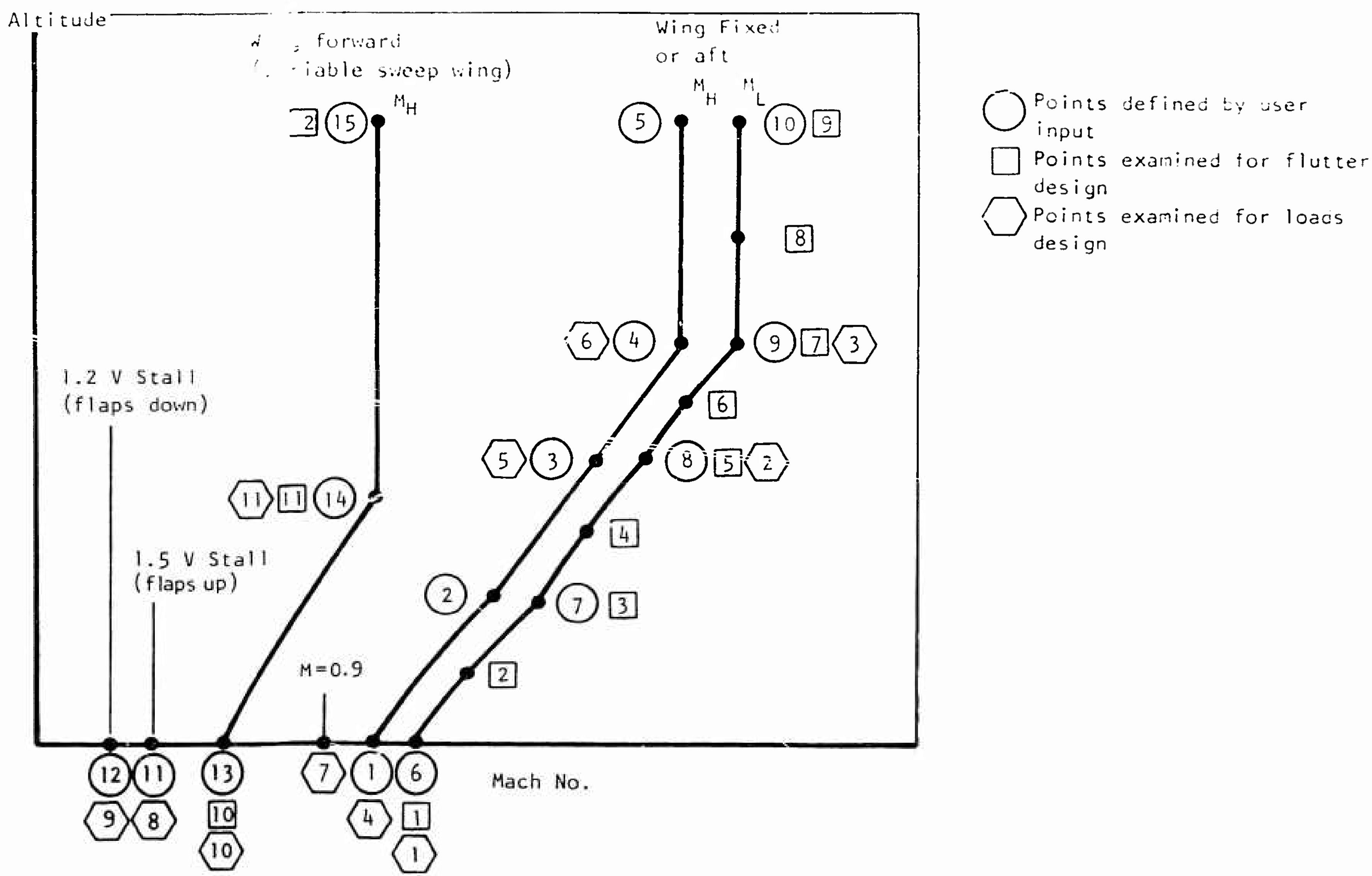



Figure 23. Speed-altitude profile points.

TABLE 8. FLUTTER EVALUATION AND DATA PROCESSING ROUTINES

| Module | Routine Name | Description | Reference Volume |
|---|---|---|---|
| Input data processing | READ | Write speed-altitude profile data in record 28 for use by air induction system weight estimation module<br><br>Store wing, horizontal tail, and vertical tail structural material identification numbers in labeled common block MISC for use in flutter and temperature module. | II |
| Data management | SPDALT | Expand input speed-altitude profile data by interpolating for intermediate points, and calculate dynamic pressure | II |
| | AVDATA | Write expanded limit speed-altitude profile data for use by fuselage weight estimation module in record 34<br><br>Write expanded limit speed-altitude profile data and horizontal tail inertial data in record 38 for use by flutter and temperature module and wing and empennage weight estimation module | II |
| Flutter and temperature | WHVCQ | Calculate design dynamic pressure and shear modulus data, and store in record 38 for use by wing and empennage eight estimation module | IV |
| Air induction system weight estimation | SPAL | Expand input speed-altitude profile data by interpolating for intermediate points, and calculate dynamic pressure | V |
| | NACELE | Determine critical flutter point, and calculate nacelle panel thickness to prevent local panel flutter | V |

TABLE 8. FLUTTER EVALUATION AND DATA PROCESSING ROUTINES (CONCL)

| Module | Routine Name | Description | Reference Volume |
| --- | --- | --- | --- |
| Wing and empennage weight estimation | CCNTL | Transfer flutter design parameters from record 38 and labeled common block MISC into working data region | VI |
| | WODATA | Transfer revised horizontal tail inertia data into record 38 for use in T-tail flutter evaluation | VI |
| | GJCAL | Evaluation control and calculations for surface and T-tail flutter | VI |
| | GJSI | Calculation of torsional stiffness to prevent surface flutter | VI |
| | GJTT | Calculation of torsional stiffness to prevent T-tail flutter | VI |
| | VFCAL | Calculation of structure thickness (metal) to satisfy required torsional stiffness. | VI |
| | ACWMS | Calculation of structure thickness (multispar composite) to satisfy required torsional stiffness | VI |
| | ACWSTR | Calculation of structure thickness (stringer composite) to satisfy required torsional stiffness. | VI |
| Fuselage weight estimation | QCRIT | Scan speed-altitude profile for critical flutter point | VII |
| | FCOVER | Calculate cover thickness to prevent local panel flutter | VII |

TABLE 9. BLANK COMMON

| Variable Name | Size | Common Location | Description |
|---|---|---|---|
| SVF | 180 | 1 | Refer to Table 12 |
| DATA | 312 | 181 | Refer to Table 11 |
| TFW | 38 | 493 | Composite flutter paramter for wing in fixed or aft position |
| TFH | 38 | 531 | Composite flutter parameter for horizontal tail |
| TFV | 38 | 569 | Composite flutter parameter for vertical tail |
| TFWF | 38 | 607 | Composite flutter parameter for wing in forward position |
| TF | 38 | 645 | Composite flutter parameter, working location |
| XMNI | 20 | 665 | Design mach number |
| Q | 20 | 685 | Incompressible dynamic pressure at design mach numbers |
| QQ | 20 | 705 | Dynamic pressure corrected for compressibility effect |
| XMATOP | 20 | 725 | Mach number associated with QQ |
| GW | 6 | 745 | Shear modulus at library temperature for wing material |
| GH | 6 | 751 | Shear modulus at library temperatures for horizontal tail material |
| GV | 6 | 757 | Shear modulus at library temperatures for vertical tail material |

TABLE 10. COMMON VARIABLES

| Variable Name | Size | Common Location | Type | Description | Subroutine Reference |
|---|---|---|---|---|---|
| ABSORP | 1 | 487 | I | 0.85, absorptivity of structure surface (DATA) | TEMPER |
| ATMOS | 1 | 489 | I | 1.0, model atmosphere indicator (DATA) | TEMPER |
| COSPHI | 1 | 488 | I | 1.0, cosine of angle from normal of (DATA) sun rays | TEMPER |
| DATA | 312 | 181 | I | Permanent data array | OLAY3, SVFTAB |
| EMISS | 1 | 485 | I | 0.85, emissivity of structure surface (DATA) | TEMPER |
| GH | 6 | 751 | C | Shear modulus at library temperatures for horizontal tail material (SVF) | OLAY3, WHVMAT, WHVQQ |
| GV | 6 | 757 | C | Shear modulus at library temperature for vertical tail material | OLAY3, WHVMAT, WHVQQ |
| GW | 6 | 745 | C | Shear modulus at library temperature for wing material | OLAY3, WHVMAT, WHVQQ |
| NHOR | 1 | 165 | C | Number of values in horizontal tail stress-G-temperature table (SVF) | WHVMAT, WHVQQ |
| NVER | 1 | 179 | C | Number of values in vertical tail stress-G-temperature table (SVF) | WHVMAT, WHVQQ |
| NWING | 1 | 151 | C | Number of values in wing stress-G-temperature table (SVF) | WHVMAT, WHVQQ |
| PSI | 23 | 1 | C | Ambient pressure at each load condition (SVF) | OLAY3 |
| Q | 20 | 685 | C | Incompressible dynamic pressure at design mach numbers | QSUB, WHVQQ |
| QQ | 20 | 705 | C | Dynamic pressure corrected for compressibility effects | QSUB, WHVQQ |

TABLE 10. COMMON VARIABLES (CONT)

| Variable Name | Size | Common Location | Type | Description | Subroutine Reference |
|---|---|---|---|---|---|
| SFLUX | 23 | 70 | C | Sun flux at each load condition (SVF) | OLAY3 |
| STH | 6 | 153 | C | Compression yield stress for horizontal tail material at library temperatures (SVF) | WHVMAT |
| STV | 6 | 167 | C | Compression yield stress for vertical tail material at library temperatures (SVF) | WHVMAT |
| STW | 6 | 70 | C | Compression yield stress for wing material at library temperatures (SVF) | WHVMAT |
| SVF | 180 | 1 | C | Major surface material properties and structure temperature data for use in airloads module | OLAY3, WHVMAT |
| SWEEP | 1 | 492 | I | 0.0, sweep of leading edge of surface (DATA) | TEMPER |
| S80H | 1 | 166 | C | Compression yield stress of horizontal tail material at 80° F (SVF) | WHVMAT |
| S80V | 1 | 180 | C | Compression yield stress of vertical tail material at 80° F (SVF) | WHVMAT |
| S80W | 1 | 152 | C | Compression yield stress of wing material at 80° F (SVF) | WHVMAT |
| TABLE | 38 | 181 | I | Mach numbers for flutter parameter tables (DATA) | QSUB |
| TAR2 | 38 | 257 | I | Flutter parameter table for aspect ratio of 2.0, quarter chord sweep of 45 degrees and taper ratio of 0.3 (DATA) | SVFTAB |
| TAR6 | 38 | 295 | I | Flutter parameter table for aspect ratio of 6.0, quarter chord sweep of 45 degrees and taper ratio of 0.3 (DATA) | SVFTAB |

TABLE 10. COMMON VARIABLES (CONT)

| Variable Name | Size | Common Location | Type | Description | Subroutine Reference |
|---|---|---|---|---|---|
| TBP | 38 | 219 | I | Flutter parameter table for aspect ratio of 4.0, quarter chord sweep of 45 degrees and taper ratio of 0.3 (DATA) | SVFTAB |
| TEMPH | 6 | 159 | C | Horizontal tail material library temperatures (SVF) | WHVMAT, WHVQQ |
| TEMPV | 6 | 173 | C | Vertical tail material library temperatures (SVF) | WHVMAT, WHVQQ |
| TEMPW | 6 | 145 | C | Wing material library temperatures (SVF) | WHVMAT, WHVQQ |
| TF | 38 | 645 | C | Composite flutter parameter, working location | QSUB, WHVQQ |
| TFH | 38 | 531 | C | Composite flutter parameter for horizontal tail | SVFTAB |
| TFV | 38 | 569 | C | Composite flutter parameter for vertical tail | SVFTAB |
| TFW | 38 | 493 | C | Composite flutter parameter for wing-fixed or aft | SVFTAB |
| TFWF | 38 | 607 | C | Composite flutter parameter for wing-forward | SVFTAB |
| TLOCAL | 23 | 47 | C | Ambient temperature for each load condition (SVF) | OLAY3 |
| TREY | 1 | 491 | I | 0.0, transition Reynolds number (DATA) | TEMPER |
| TSB0 | 38 | 333 | I | Flutter parameter for aspect ratio of 4.0, quarter chord sweep of 0 degrees and taper ratio of 0.3 (DATA) | SVFTAB |
| TSB60 | 38 | 371 | I | Flutter parameter for aspect ratio of 4.0, quarter chord sweep of 60 degrees and taper ratio of 0.3 (DATA) | SVFTAB |
| TSKINF | 23 | 116 | C | Equilibrium skin temperature at each load condition, ° F (SVF) | OLAY3 |

TABLE 10. COMMON VARIABLES (CONCL)

| Variable Name | Size | Common Location | Type | Description | Subroutine Reference |
|---|---|---|---|---|---|
| TSKINR | 23 | 93 | C | Equilibrium skin temperature at each load condition, °R (SVF) | OLAY3 |
| TTOTAL | 23 | 47 | C | Total temperature at each load condition, °R (SVF) | OLAY3 |
| TTR0 | 38 | 409 | I | Flutter parameter for aspect ratio of 4.0, quarter chord sweep of 45 degrees and taper ratio of 0.0 (DATA) | SVFTAB |
| TTR60 | 38 | 447 | I | Flutter parameter for aspect ratio of 4.0, quarter chord sweep of 45 degrees and taper ratio of 0.6 (DATA) | SVFTAB |
| VFF | 1 | 485 | I | 1.15, flutter margin (DATA) | WHVQQ |
| XLNGTH | 1 | 490 | I | 10.0, distance from front of body to temperature point (DATA) | TEMPER |
| XMATOP | 20 | 725 | C | Mach numbers associated with QQ | QSUB, WHVQQ |
| XMNI | 2 | 665 | C | Design mach numbers | QSUB, WHVQQ |

TABLE 11. DATA ARRAY VARIABLES

| Loc | Variable Name | Description | Subroutine Reference |
|---|---|---|---|
| 1<br>.<br>38 | TABLE (1)<br>.<br>TABLE (38) | * Mach numbers for flutter parameter curves | SVFTAB |
| 39<br>.<br>76 | TBP (1)<br>.<br>TBP (38) | * Flutter parameter for aspect ratio of 4.0, sweep of the quarter chord of 45°, and taper ratio of 0.3 | SVFTAB |
| 77<br>.<br>114 | TAR2 (1)<br>.<br>TAR2 (38) | * Flutter parameter for aspect ratio of 2.0, sweep of the quarter chord of 45°, and taper ratio of 0.3 | SVFTAB |
| 115<br>.<br>152 | TAR6 (1)<br>.<br>TAR6 (38) | * Flutter parameter for aspect ratio of 6.0, sweep of the quarter chord of 45°, and taper ratio of 0.3 | SVFTAB |
| 153<br>.<br>190 | TSB0 (1)<br>.<br>TSB0 (38) | * Flutter parameter for aspect ratio of 4.0, sweep of the quarter chord of 0°, and taper ratio of 0.3 | SVFTAB |
| 191<br>.<br>228 | TSB60 (1)<br>.<br>TSB60 (38) | * Flutter parameter for aspect ratio of 4.0, sweep of the quarter chord of 60°, and taper ratio of 0.3 | SVFTAB |
| 229<br>.<br>266 | TTR0 (1)<br>.<br>TTR0 (38) | * Flutter parameter for aspect ratio of 4.0, sweep of the quarter chord of 45°, and taper ratio of 0 | SVFTAB |
| 267<br>.<br>304 | TTR60 (1)<br>.<br>TTR60 (38) | * Flutter parameter for aspect ratio of 4.0, sweep of the quarter chord of 45°, and taper ratio of 0.6 | SVFTAB |
| 305 | VFF | 1.15, Flutter margin | WHVQQ |
| 306 | EMISS | 0.85, emissivity of structure surface | TEMPER |
| 307 | ABSORP | 0.85, absorptivity of structure surface | TEMPER |
| 308 | COSPHI | 1.0, cosine of angle from normal of suns rays | TEMPER |
| 309 | ATMOS | 1.0, model atmosphere indicator, 1.0 = 1962 U.S. standard | TEMPER |
| 310 | XLNGTH | 10.0, distance from front of surface to temperature calculation point | TEMPER |
| 311 | TREY | 0.0, transition Reynolds number | TEMPER |
| 312 | SWEEP | 0.0, sweep of leading edge of surface | TEMPER |

*Values for this array can be found in Table 7 in the flutter methodology section.

TABLE 12. SVF ARRAY VARIABLES

| Loc | Variable Name | Description |
|---|---|---|
| 1 | PSI (1) | Ambient pressure at load condition 1, psi |
| . | . | to |
| 23 | PSI (23) | Ambient pressure at load condition 23, psi |
| 24 | TLOCAL (1) | Ambient temperature at load condition 1, °R |
| . | . | to |
| 46 | TLOCAL (23) | Ambient temperature at load condition 23, °R |
| 47 | TTOTAL (1) | Total temperature at load condition 1, °R |
| . | . | to |
| 69 | TTOTAL (23) | Total temperature at load condition 23, °R |
| 70 | SFLUX (1) | Sun flux at load condition 1, BTU/hr/ft$^2$ |
| . | . | to |
| 92 | SFLUX (23) | Sun flux at load condition 23, BTU/hr/ft$^2$ |
| 93 | TSKINR (1) | Equilibrium skin temperature at load condition 1, °R |
| . | . | to |
| 115 | TSKINR (23) | Equilibrium skin temperature at load condition 23, °R |
| 116 | TSKINF (1) | Equilibrium skin temperature at load condition 1, °F |
| . | . | to |
| 138 | TSKINF (23) | Equilibrium skin temperature at load condition 23, °F |
| 139 | STW (1) | Wing material compression yield stress at library temperature 1, psi |
| . | . | to |
| 144 | STW (6) | Wing material compression yield stress at library temperature 6, psi |
| 145 | TEMPW (1) | Wing material library temperature 1, °F |
| . | . | to |
| 150 | TEMPW (6) | Wing material library temperature 6, °F |
| 151 | NWING | Number of values in wing stress-G-temperature table |
| 152 | S80W | Wing material compression yield stress at 80° F |
| 153 | STH (1) | Horizontal tail material compression yield stress at library temperature 1, psi |
| . | . | |

TABLE 12. SVF ARRAY VARIABLES (CONCL)

| Loc | Variable Name | Description |
|---|---|---|
| 158 | STH (6) | Horizontal tail material compression yield stress at library temperature 6, psi |
| 159<br>.<br>164 | TEMPH (1)<br>.<br>TEMPH (6) | Horizontal tail material library temperature 1, °F<br>to<br>Horizontal tail material library temperature 6, °F |
| 165 | NHOR | Number of values in horizontal tail stress-G-temperature table |
| 166 | S80H | Horizontal tail material compression yield stress at 80° F |
| 167<br>.<br>172 | STV (1)<br>.<br>STV (6) | Vertical tail material compression yield stress at library temperature 1, psi<br>to<br>Vertical tail material compression yield stress at library temperature 6, psi |
| 173<br>.<br>178 | TEMPV (1)<br>.<br>TEMPV (6) | Vertical tail material library temperature 1, °F<br>to<br>Vertical tail material library temperature 6, °F |
| 179 | NVER | Number of values in vertical tail stress-G-temperature table |
| 180 | S80V | Vertical tail material compression yield stress at 80° F |

TABLE 13. ITEMS USED FROM BC ARRAY

| BC Loc | Description |
|---|---|
| 19 | Altitude at point 1 on speed profile with wings fixed or aft, ft |
| 20 | Altitude at point 2 on speed profile with wings fixed or aft, ft |
| 21 | Altitude at point 3 on speed profile with wings fixed or aft |
| 22 | Level-flight maximum speed at altitude 1 with wings fixed or aft, mach No. |
| 23 | Level-flight maximum speed at altitude 2 with wings fixed or aft, mach No. |
| 25 | Altitude at point 1 on speed profile with wings forward (variable sweep only), ft |
| 26 | Altitude at point 2 on speed profile with wings forward (variable sweep only), ft |
| 28 | Level-flight maximum speed at altitude 1 with wings forward, mach No. |
| 29 | Level-flight maximum speed at altitude 2 with wings forward, mach No. |
| 31 | Minimum speed with flaps up at maximum design weight, knots |
| 32 | Minimum speed with flaps down at landing desing weight, knots |
| 166 | Limit speed at altitude 1 with wings fixed or aft, mach No. |
| 167 | Limit speed at altitude 2 with wings fixed or aft, mach No. |
| 168 | Limit speed at altitude 3 with wings fixed or aft, mach No. |

TABLE 14. XMISC ARRAY VARIABLES (MISC BLOCK)

| Loc | Description | Subroutine Reference |
|---|---|---|
| 1 . 4 | Controls and design data used by other program modules | |
| 5 | Dynamic pressure for wing design, $lb/ft^2$ (wing fixed or aft) | WHVQQ |
| 6 | Dynamic pressure for horizontal tail design, $lb/ft^2$ | WVHQQ |
| 7 | Dynamic pressure for vertical tail design, $lb/ft^2$ | WHVQQ |
| 8 . 11 | Design temperature data used in airloads module | |
| 12 | Aspect ratio of wing (fixed or aft) | SVFTAB |
| 13 | Quarter chord sweep of wing (fixed or aft), deg | SVFTAB |
| 14 | Taper ratio of wing (fixed or aft) | SVFTAB |
| 15 | Wing structural material identification number | WHVMAT, WHVQQ |
| 16 | Aspect ratio of horizontal tail | SVFTAB |
| 17 | Quarter chord sweep of horizontal tail, deg | SVFTAB |
| 18 | Taper ratio of horizontal tail | SVFTAB |
| 19 | Horizontal tail structural material identification number | WHVMAT, WHVQQ |
| 20 | Aspect ratio of vertical tail | SVFTAB |
| 21 | Quarter chord sweep of vertical tail, deg | SVFTAB |
| 22 | Taper ratio of vertical tail | SVFTAB |
| 23 | Vertical tail structural material identification number | WHVMAT, WHVQQ |

TABLE 14. XMISC ARRAY VARIABLES (MISC BLOCK) (CONCL)

| Loc | Description | Subroutine Reference |
|---|---|---|
| 24 | Weight data used by other program modules | |
| 25 | Aspect ratio of wing (forward) | SVFTAB |
| 26 | Quarter chord sweep of wing (forward), deg | SVFTAB |
| 27 | Taper ratio of wing (forward) | SVFTAB |
| 28 | Wing structural material shear modulus at design flutter point, $lb/in.^2$ | WHVQQ |
| 29 | Horizontal tail structural material shear modulus at design flutter point, $lb/in.^2$ | WHVQQ |
| 30 | Vertical tail structural material shear modulus at design flutter point, $lb/in.^2$ | WHVQQ |
| 31<br>.<br>52 | Controls and design data used by other program modules | |
| 53 | Vertical tail-type indicator<br>-1 = single tail<br>0 = dual tail<br>1 = T-tail | WHVQQ |
| 54<br>.<br>100 | Controls and design data used by other program modules | |

TABLE 15. MASS STORAGE FILE RECORDS

| Record No. | Variable & Length | Write Routine | Read Routine | Description |
|---|---|---|---|---|
| 12 | DATA (312) | Input data processing module | OLAY3 | Flutter parameter tables and temperature parameters (refer to Table 11) |
| 22 | EC (195) | Data management module | OLAY3 | Vehicle design data (refer to Table 13) |
| 31 | SVF (180) | OLAY3 | Airloads module | Wing, horizontal tail, and vertical tail material property and structure temperature data (refer to Table 12) |
| 37 | GJDAT (100) | Input data processing module | WHVQQ | T-tail flutter parameter tables (refer to Table 18) |
| 38 | SPAL (50) | Data management module, WHVQQ | WHVQQ | Speed-altitude profile and T-tail inertia data (refer to Table 17) |
| 41-60 | PM (300) | Input data processing module | WHVMAT | Material property library data (refer to subroutine description, WHVMAT) |

SUBROUTINE DESCRIPTIONS

PROGRAM OLAY3

General Description

Deck name: OLAY3
Entry name: CALL OVERLAY (5HALPHA, 3, 0)
Called by: OLAY00
Subroutines called: TEMPER, WHVQQ

This is the control program for the flutter and temperature module, and it preforms the basic initialization tasks. Arrays to be used for output are cleared. Required input is read from the following files:

- File 12, the DATA array, containing permanent data for both flutter and temperature (refer to Table 11)

- File 22, the BC array, providing the altitude and mach number data for the load conditions (refer to Table 13)

For each load condition, the program sets up altitude and mach number as shown in Table 16 and Figure 23. Subroutine TEMPER is called to determine the pressure and temperature data required by the airloads module, for each load condition. These arrays are saved in SVF and are described in Table 12.

After the load condition loop is completed, subroutine WHVQQ is called to determine material related and flutter design items.

The SVF array is saved in file 31.

Arrays and Variables Used

BC array items are shown in Table 13.

Arrays and Variables Calculated

PSI, SFLUX, TLOCAL, TSKINF, TSKINR, and TTOTAL are subarrays of SVF and are described in Table 12.

TABLE 16. ALTITUDE AND MACH NUMBER FOR EACH FLIGHT LOAD CONDITION

| Load Condition No. | Speed-Altitude Profile Point (Figure 23) | Altitude, ft (ALT Array) | Mach No. (XMACH Array) |
|---|---|---|---|
| 1 | 1 | BC(19) | BC(166) |
| 2 | 2 | BC(20) | BC(167) |
| 3 | 3 | BC(21) | BC(168) |
| 4 | 7 | 0.0 | 0.90 |
| 5 | 10 | BC(25) | BC(28) |
| 6 | 4 | BC(19) | BC(22) |
| 7 | 10 | BC(25) | BC(28) |
| 8 | 8 | 0.0 | (1.5/661.3) BC(31) |
| 9 | 9 | 0.0 | (1.2/661.3) BC(32) |
| 10 | 4 | BC(19) | BC(22) |
| 11 | 5 | BC(20) | BC(23) |
| 12 | 10 | BC(25) | BC(28) |
| 13 | 11 | BC(26) | BC(29) |
| 14 | 4 | BC(19) | BC(22) |
| 15 | 5 | BC(20) | BC(23) |
| 16 | 10 | BC(25) | BC(28) |
| 17 | 11 | BC(26) | BC(29) |
| 18 | 4 | BC(19) | BC(22) |
| 19 | 5 | BC(20) | BC(23) |
| 20 | 1 | BC(19) | BC(166) |
| 21 | 3 | BC(21) | BC(168) |
| 22 | 1 | BC(19) | BC(166) |
| 23 | 3 | BC(21) | BC(168) |

Refer to Table 13 for description of BC array variables.

## Scratch Arrays and Variables

- ALT, XMACH described in Table 16
- BB (9), BB (10) intermediate storage for altitude and mach number
- IER error indicator returned by TEMPER

## Labeled Common Arrays

None

## Mass Storage Records Used

- Reads: record 12, DATA, Table 11
  record 22, PC, Table 13
- Write: record 31, SVF, Table 12

## Error Messages

This one-line message is printed if IER is greater than 15, which indicates that the skin temperature calculation failed to converge.

LOAD CONDITION XX - THE TEMPERATURE LOOP DID NOT CLOSE IN 100 ITERATIONS.
THE SKIN TEMPERATURE YYY.YYY IS FROM THE LAST ITERATION.

# FUNCTION HBL

## General Description

| | |
|---|---|
| Deck name: | HBL |
| Entry name: | HBL (XMACH, PO, TO, TSK, DIST, XLAMDA, REY, METHOD, IER, TOTI, TAWI) |
| Called by: | TEMPER |
| Subroutines called: | None |

This routine calculates boundary layer heat transfer coefficient. Although the routine has several calculation choices, it is used here only for flat plate (wedges or cylinders) alined with the flow. This choice is with METHOD equal to 1, which is set in TEMPER.

## Arrays and Variables Used

The input arguments are:

| | |
|---|---|
| DIST | Characteristic length - METHOD = 1 uses distance aft of leading edge, XLNGTH, which is set to 10.0. |
| METHOD | Calculation choice - TEMPER sets to 1. |
| PO | Local static pressure - TEMPER sets to PRESS, which is the result of function PRESH. |
| REY | Transition Reynolds number - TEMPER sets to TREY, which was set to 1,000,000.0 by TBL. |
| TAWI | Adiabatic wall temperature - set to BLT, the result of function TBL. |
| TOTI | Total temperature - set to TOTT, the result of function TTO. |
| TSK | Skin temperature - set to XSKIN, the estimated skin temperature in the iteration loop in TEMPER. |
| TO | Local static temperature - set to TLOC, the result of function TEMALT. |
| XLAMDA | Angle of sweep - set to SWEEP, which is set to zero. |
| XMACH | Mach number. |

## Arrays and Variables Calculated

| | |
|---|---|
| HBL | The boundary layer heat transfer coefficient is returned to the expression using the function. |
| IER | The error indicator is set to zero when conditions are met, to values from 1 through 8 for various errors, but is not tested on return. |

## Scratch Variables

| | |
|---|---|
| ANAL | Prandtl number raised to -2/3, factor in laminar heat transfer coefficient equation |
| ANUM | Factor in turbulent heat transfer coefficient equation |
| ASQ | Factor in turbulent heat transfer coefficient |
| B | Partial term in turbulent flow equations |
| COND | Conductivity at reference temperature, TSTR |
| CSFI | Counter in iteration loop for turbulent skin friction - maximum number trials is 101, then the last value is used. |
| CSFN | Skin friction coefficient |
| CSFNI | Initial and previous skin friction coefficient for turbulent flow |
| EXPO | Partial term in values for static temperature |
| EXP1 | Partial term in values for reference temperature |

| | |
|---|---|
| GAMMAO | Ratio of specific heats at static temperature |
| GRHO | $g\rho$ - density at reference temperature |
| GRHOO | $g\rho$ - density at static temperature |
| GVIS | $g\mu$ - viscosity at reference temperature |
| GVISO | $g\mu$ - viscosity at static temperature |
| PRN | Prandtl number |
| PSI | Term in turbulent flow equation for skin friction |
| REN | Reynolds number based on reference temperature |
| RENO | Reynolds number based on static temperature |
| SPHT | Specific heat at constant pressure at reference temperature |
| SPHTO | Specific heat at constant pressure at static temperature |
| SPSDO | Speed of sound at static temperature |
| TAW | Adiabatic wall temperature |
| THETA | Term in turbulent flow equation for skin friction |
| TOT | Total temperature set from TOTI |
| TSTR | Reference temperature - weighted average through the boundary layer |
| VELO | Vehicle speed |

Labeled Common Arrays

None

Mass Storage File Records Used

None

Error Messages

None

FUNCTION PRESH

General Description

Deck name: PRESH
Entry name: PRESH (ALT)
Called by: TEMPER, WHVQQ
Subroutine called: None

This routine returns pressure for the input geopotential altitude using the relation defined by the U.S. standard atmosphere, 1962.

## Arrays and Variables Used

The input argument ALT is th geopotential altitude.

## Arrays and Variables Calculated

PRESH The resulting pressure is returned directly to the expression where the function is used

## Scratch Variables

| | |
|---|---|
| ALOFT | Altitude divided by 1,000.0, a factor in the pressure equations |
| ZEXPO | Exponential term for altitudes between 36,089.239 and 65,616.80 |

## Labeled Common Arrays

None

## Mass Storage File Records Used

None

## Error Messages

This message is printed for ALT greater than 154,199.48:

WARNING - ALTITUDE IS BEYOND VALID RANGE OF PRESH.

The pressure returned is calculated by the equation for the highest altitude range.

## SUBROUTINE QINC

## General Description

Deck name: QINC
Entry name: QINC (XM, PO, TO, Q)
Called by: WHVQQ
Subroutines called: None

This routine calculates incompressible dynamic pressure for a given mach number, local pressure, and local temperature.

Arrays and Variables Used

| | |
|---|---|
| PO | Input argument, ambient pressure, $lb/ft^2$ |
| TO | Input argument, ambient temperature, $^\circ$ R |
| XM | Mach number |

Arrays and Variables Calculated

| | |
|---|---|
| Q | Output argument, incompressible dynamic pressure |

Scratch Arrays and Variables

| | |
|---|---|
| EXP1 | $e^{TOT}$, intermediate term |
| G | 32.17405, acceleration of gravity, $ft/sec^2$ |
| GAMMA | Ratio of specific heats |
| R | 53.35045, gas constant, ft/$^\circ$ R |
| RHO | Density of air, $slug/ft^3$ |
| SPHT | Specific heat at constant pressure, ft-lb/lb/$^\circ$ R |
| SPSD | Speed of sound, ft/sec |
| TOT | 5526.0/TO, intermediate term |

Labeled Common Arrays

None

Mass Storage File Records Used

None

Error Messages

None

SUBROUTINE QSUB

## General Description

Deck name: QSUB
Entry name: QSUB (LIM)
Called by: WHVQQ
Subroutines called: None

This routine calculates the design dynamic pressure corrected for compressibility effects at each point on the speed-altitude profile. The argument, LIM, is the number of points to be used. The other required data and the results are transmitted via blank common.

## Arrays and Variables Used

| | |
|---|---|
| LIM | Input argument, number of mach number-dynamic pressure points |
| Q | Incompressible dynamic pressure at design mach numbers |
| TABLE | Mach numbers for flutter parameter arrays |
| TF | Flutter parameter array for the surface being evaluated |
| XMNI | Design mach numbers |

## Arrays and Variables Calculated

| | |
|---|---|
| QQ | Dynamic pressure corrected for compressibility effects |
| XMATOP | Mach numbers associated with QQ |

## Scratch Arrays and Variables

| | |
|---|---|
| I | General index, range from 1 to LIM |
| N | Index for search of TABLE |
| NS | Index of highest TF from 1 to NSAVE inclusive |
| NSAVE | Index of TABLE mach no, just below or equal to XMN |
| TFANS | Set to larger of TFP or TOP |
| TFP | Composite flutter parameter at XMN |
| TOP | Highest TF from 1 to NSAVE inclusive |
| XMN | Mach number from XMNI in work |

## Labeled Common Arrays

None

## Mass Storage File Records Used

None

## Error Messages

None

## FUNCTION SOLARG

## General Discussion

| | |
|---|---|
| Deck name: | SOLARG |
| Entry name: | SOLARG (ALT) |
| Called by: | TEMPER |
| Subroutines called: | None |

This routine gives solar flux, as a function of altitude, in British thermal units per hour per square foot through a plane perpendicular to the sun's rays.

For altitudes less than 30,000.0 feet:

$$SOLARG = 435.0 - 0.141 \left(\frac{30{,}000.0 - ALT}{1{,}000.0}\right)^{1.85}$$

For altitudes of 30,000.0 feet and over:

$$SOLARG = 435.0$$

## Arrays and Variables Used

ALT Input geopotential altitude

## Arrays and Variables Calculated

SOLARG Resultant solar flux is returned to the using expression

Scratch Variables

ALOFT   Altitude divided by 1,000.0

Labeled Common Arrays

None

Mass Storage File Records Used

None

Error Messages

None

SUBROUTINE SVFTAB

General Description

Deck name: SVFTAB
Entry name: SVFTAB
Called by: WHVQQ
Subroutines called: None

This routine calculates the composite flutter parameters. These tables are set up for those surfaces for which a nonzero value of aspect ratio is given. Calculations can be made for the wing fixed or aft, wing forward, horizontal tail, and/or vertical tail. The calculated arrays will be printed if IP(41) is zero. (See Figure 24.)

Arrays and Variables Used

| | |
|---|---|
| TABLE | Mach numbers for the flutter parameter curves |
| TAR2 | Flutter parameter curve for aspect ratio of 2.0, sweep of the quarter chord of 45°, and taper ratio of 0.3. |
| TAR6 | Flutter parameter array for aspect ratio of 6.0, sweep of the quarter chord of 45°, and taper ratio of 0.3 |
| TEP | Flutter parameter array for aspect ratio of 4.0, sweep of the quarter chord of 45°, and taper ratio of 0.3 |

*** WING (FIXED OR AFT) ***  ** SVFTAB - IP(41) **

FLUTTER PARAMETER VS MACH NUMBER

| MACH NO. | AR = 1.68 | SWEEP(C/4) = 70.1 DEG | TAPER = .300 | COMPOSITE |
|---|---|---|---|---|
| 0.200 | .0201 | .0817 | .0600 | .0274 |
| 0.400 | .0402 | .1634 | .1200 | .0548 |
| 0.600 | .0604 | .2452 | .1800 | .0822 |
| 0.800 | .0805 | .3269 | .2400 | .1096 |
| 0.825 | .0830 | .3419 | .2550 | .1113 |
| 0.850 | .0855 | .3585 | .2750 | .1115 |
| 0.875 | .0880 | .3752 | .2950 | .1120 |
| 0.900 | .0906 | .3935 | .3100 | .1149 |
| 0.925 | .0931 | .4122 | .3220 | .1191 |
| 0.950 | .0947 | .4323 | .3270 | .1252 |
| 0.975 | .0964 | .4486 | .3300 | .1311 |
| 1.000 | .0981 | .4623 | .3320 | .1366 |
| 1.025 | .0998 | .4733 | .3330 | .1418 |
| 1.050 | .1006 | .4824 | .3320 | .1462 |
| 1.075 | .1014 | .4877 | .3290 | .1504 |
| 1.100 | .1023 | .4921 | .3250 | .1549 |
| 1.125 | .1031 | .4921 | .3200 | .1586 |
| 1.150 | .1027 | .4911 | .3140 | .1600 |
| 1.175 | .1014 | .4891 | .3070 | .1616 |
| 1.200 | .1006 | .4838 | .3000 | .1622 |
| 1.300 | .0981 | .4518 | .2780 | .1594 |
| 1.400 | .0956 | .4124 | .2670 | .1476 |
| 1.500 | .0939 | .3873 | .2620 | .1388 |
| 1.600 | .0931 | .3776 | .2590 | .1357 |
| 1.700 | .0922 | .3740 | .2570 | .1342 |
| 1.800 | .0922 | .3706 | .2570 | .1330 |
| 1.900 | .0931 | .3733 | .2580 | .1346 |
| 2.000 | .0939 | .3770 | .2600 | .1361 |
| 2.100 | .0956 | .3806 | .2620 | .1388 |
| 2.200 | .0973 | .3860 | .2640 | .1422 |
| 2.300 | .0989 | .3913 | .2660 | .1455 |
| 2.400 | .1006 | .3960 | .2690 | .1481 |
| 2.500 | .1023 | .4006 | .2720 | .1507 |
| 3.000 | .1090 | .4257 | .2920 | .1589 |
| 3.500 | .1165 | .4520 | .3150 | .1672 |
| 4.000 | .1249 | .4793 | .3390 | .1766 |
| 4.500 | .1341 | .5094 | .3640 | .1877 |
| 5.000 | .1442 | .5404 | .3900 | .1998 |

Figure 24. Sample output - flutter parameter versus mach number.

| | |
|---|---|
| TSR0 | Flutter parameter array for aspect ratio of 4.0, sweep of the quarter chord of 0°, and taper ratio of 0.3 |
| TSR60 | Flutter parameter array for aspect ratio of 4.0, sweep of the quarter chord of 60°, and taper ratio of 0.3 |
| TTR0 | Flutter parameter array for aspect ratio of 4.0, sweep of the quarter chord of 45°, and taper ratio of 0.0 |
| TTR60 | Flutter parameter array for aspect ratio of 4.0, sweep of the quarter chord of 45°, and taper ratio of 0.6 |

Arrays and Variables Calculated

| | |
|---|---|
| TFH | Flutter parameter array for horizontal tail |
| TFV | Flutter parameter array for vertical tail |
| TFW | Flutter parameter array for wing (fixed or aft) |
| TFWF | Flutter parameter array for wing (forward) |

Scratch Arrays and Variables

Geometry items relocated from XMISC:

| Name | XMISC Location | Description |
|---|---|---|
| ARH | 16 | Aspect ratio of horizontal tail |
| ARV | 20 | Aspect ratio of vertical tail |
| ARW | 12 | Aspect ratio of wing (fixed or aft) |
| ARWF | 25 | Aspect ratio of wing (forward) |
| SWH | 17 | Quarter chord sweep of horizontal tail |
| SWV | 21 | Quarter chord sweep of vertical tail |
| SWW | 13 | Quarter chord sweep of wing (fixed or aft) |
| SWWF | 26 | Quarter chord sweep of wing (forward) |
| TRH | 18 | Taper ratio of horizontal tail |
| TRV | 22 | Taper ratio of vertical tail |
| TRW | 14 | Taper ratio of wing (fixed or aft) |
| TRWF | 27 | Taper ratio of wing (forward) |

Other scratch items:

| Name | Size | Description |
|---|---|---|
| AR | 1 | Aspect ratio, working location |
| I | 1 | Scratch counter, indicates which surface in work |
| N | 1 | Scratch counter |
| PCNT | 1 | Factor to modify input flutter parameter array for the required geometry |

| Name | Size | Description |
|---|---|---|
| SB | 1 | Quarter chord sweep, working location |
| TAR | 38 | Flutter parameter interpolated to the aspect ratio in work |
| TF | 38 | Composite flutter parameter, working location |
| TR | 1 | Taper ratio, working location |
| TSB | 38 | Flutter parameter interpolated to the quarter chord sweep in work |
| TTR | 38 | Flutter parameter interpolated to the taper ratio in work |

Labeled Common Arrays

IP(41) from block IPRINT is the print/no-print indicator. If IP(41) is zero, the tables of mach number versus flutter parameter are printed.

The XMISC items used from block MISC are itemized under geometry items in the foregoing description of scratch.

Mass Storage File Records Used

None

Error Messages

None

FUNCTION TBL

General Description

| | |
|---|---|
| Deck name: | TBL |
| Entry name: | TBL (XMACH, PO, TO, TSK, DIST, XLAMDA, REY, METHOD, IER, TOT1) |
| Called by: | TEMPER |
| Subroutines called: | None |

This routine calculates adiabatic wall temperature. In the calculation, variation of specific heat, viscosity, conductivity, and density are taken into account. Since these variables are interrelated, an iterative solution is used.

## Arrays and Variables Used

The input arguments are:

| | |
|---|---|
| DIST | Distance aft of the leading edge for which XLNGTH is used. Its value is 10.0. |
| METHOD | Calculated choice 0 or 1 returns skin temperature; 2 returns stagnation temperature. In TEMPER, use METHO, which is 1. |
| PO | Local static pressure - use PRESS, which is the result of function PRESH. |
| REY | Transition Reynolds number - use TREY, which is set to zero, indicating that the routine uses 1,000,000.0 <br><br> Note REY is reset to 1,000,000.0. |
| TOTT | Total temperature - use TOTT, the result of function TTO. |
| TSK | Skin temperature - use XSKIN, the estimated skin temperature in the iteration loop in TEMPER. |
| TO | Local static temperature - use TLOC, the result of function TEMALT. |
| XLAMDA | Angle of sweep - use SWEEP, which is set to zero. |
| CMACH | Local mach number. |

## Arrays and Variables Calculated

| | |
|---|---|
| TEL | Adiabatic wall temperature is returned to the using expression. |
| IER | The error indicator - if two, the solution failed to converge, and the value returned is based on a recovery factor of 0.875. |
| REY | If input as zero, is reset to 1,000,000.0. |

## Scratch Variables

| | |
|---|---|
| COND | Thermal conductivity of boundary layer |
| EXP1 | Term in specific heat equations |
| GAMMA | Ratio of specific heats |
| GRHO | Specific weight, gr, $lb/ft^3$ |
| GVIS | Viscosity times gravitational acceleration |
| IND | Counter in iteration loop |
| PRN | Prandtl number |
| RECOV | Recovery factor |
| REN | Freestream Reynolds number |
| SPHT | Specific heat at constant pressure |
| SPHTV | Specific heat at constant volume |

| | |
|---|---|
| SPSD | Speed of sound |
| TBLI | Initial and prior boundary layer temperature |
| TSTR | Reference temperature |
| TOT | Total temperature, set to TOTI |
| VEL | Velocity of air vehicle |

Labeled Common Arrays

None

Mass Storage File Records Used

None

Error Messages

None

FUNCTION TEMALT

General Description

| | |
|---|---|
| Deck name: | TEMALT |
| Entry name: | TEMALT (ALT, TI) |
| Called by: | TEMPER, WHVQQ |
| Subroutines called: | None |

This routine returns air temperature for an input pressure altitude and model atmosphere number. The altitude is assumed to be in terms of the 1962 U.S. standard atmosphere geopotential altitude. There are five model atmospheres plus the defrost day of MIL-T-5842 in this routine.

| TI | Atmosphere |
|---|---|
| 1.0 | U.S. Standard atmosphere, 1962 |
| 2.0 | Cold atmosphere as given in Table II of MIL-STD-210A |
| 3.0 | Hot atmosphere as given in Table III of MIL-STD-210A |
| 4.0 | Polar atmosphere as given in Table IV of MIL-STD-210A |
| 5.0 | Tropical atmosphere as given in Table V of MIL-STD-210A |
| 6.0 | Defrost day of MIL-T-5842 |

Model atmosphere number one is specified in permanent data, but the coding for the other atmospheres is complete and intact.

## Arrays and Variables Used

The input arguments are:

| | |
|---|---|
| ALT | Geopotential altitude |
| II | ATMOS, DATA(309) |

## Arrays and Variables Calculated

TEMALT The resulting temperature, is returned directly to the expression using the function

## Scratch Variables

| | |
|---|---|
| ALOFT | Altitude divided by 1,000.0 |
| II | Model atmosphere number |
| SLOPE | Intermediate calculation term |

## Labeled Common Arrays

None

## Mass Storage File Records Used

None

## Error Messages

WARNING - ALTITUDE IS BEYOND VALID RANGE OF TEMALT.

This message is printed if the altitude is above the highest altitude for the model atmosphere in use. For the No. 1.0 atmosphere, the altitude is 154,200.0 feet and the temperature returned is calculated by the equation for altitude range 104,987.0 to 154,200.0.

SUBROUTINE TEMPER

## General Description

Deck name: TEMPER

Entry name: TEMPER (XMACH, ALT, PRESSI, TLOC, TTOT, SUN, XSKIN, XSKINF, IER)

Called by: OLAY3, WHVQQ

Function routines used: PRESH, TEMALT, TTO, SOLARG, TBL, HBL, TSKIN

This routine determines the pressure, local temperature, total temperature, solar flux, and skin temperature for a given altitude and mach number. In the iteration for skin temperature, if the loop does not close to 0.1 degree in 90 iterations, the tolerance is loosened to 1 degree. If the calculation does not converge in 10 more iterations, the error indicator is set and the last temperature is returned.

## Arrays and Variables Used

| | |
|---|---|
| ABSORP | Structure surface absorptivity |
| ALT | Input argument altitude |
| ATMOS | Model atmosphere identification number |
| COSPHI | Cosine of solar angle of incidence |
| EMISS | Structure surface emissivity |
| SWEEP | Sweep of surface |
| TREY | Transition Reynolds number |
| XLNGTH | Distance from front of body to point of temperature calculation |
| XMACH | Input argument mach number |

## Arrays and Variables Calculated

The output arguments are:

| | |
|---|---|
| IER | Error indicator set to 16 if skin temperature calculation does not converge |
| PRESSI | Pressure at altitude, lb/in.$^2$ |
| SUN | Solar flux in BTU/hr/sq ft |
| TLOC | Temperature at altitude, $^{\circ}$ R |
| TTOT | Temperature of moving vehicle's immediate surroundings, $^{\circ}$ R |
| XSKIN | Skin temperature (assumed value during iteration) $^{\circ}$ R |
| XSKINF | Skin temperature, $^{\circ}$ F |

## Scratch Variables

| | |
|---|---|
| A | Term in temperature equation |
| B | Coefficient in temperature equation |
| BLHTC | Boundary layer heat transfer coefficient |
| BLT | Boundary layer temperature |
| CRIT | Tolerance in skin temperature calculation set to 0.1 for first 90 passes, and 1.0 for last 10. |
| IPASS | Pass counter in iteration loop |
| IPTEN | IPASS/10 used in testing |
| METHO | 1, used for TBL for flow parallel to the surface |
| METHOD | 1, used for HBL for parallel plane flow |
| PRESS | Pressure at altitude, $lb/ft^2$ |
| XSKIN2 | Second skin temperature in iteration, the new temperature which is placed in XSKIN each time to use as the estimate for the next pass |
| XY | Absolute value of (XSKIN2-XSKIN), for testing against CRIT |

## Labeled Common Arrays

None

## Mass Storage File Records Used

None

## Error Messages

None

# FUNCTION TSKIN

## General Description

| | |
|---|---|
| Deck name: | TSKIN |
| Entry name: | TSKIN (A, B, C, IER) |
| Called by: | TEMPER |
| Subroutines called: | None |

This routine solves the flux equation for the positive real root of skin temperature by an iterative process. The process used is to find the intersection in the first quadrant of the lines $Y = X$ and $Y = A - B X^4$.

## Arrays and Variables Used

Input arguments A and B are calculated in TEMPER. Argument C, the iteration tolerance, is set to CRIT and is 0.1 or 1.0, depending on how many passes have been made in the overall loop which uses this function.

## Arrays and Variables Calculated

TSKIN, the resulting temperature, is returned to the expression using the function.

IER is set to 16 if this function does not converge in 51 calculation passes.

## Scratch Variables

| | |
|---|---|
| DENOM | Denominator of new X (i.e., XS) in iteration loop |
| ICOUNT | Counter in iteration loop maximum is 51 |
| XP | Starting value is $\left(\frac{A}{B}\right)^{1/4}$, and then set from XR |
| XR | X where the secant of the curve $Y = A - B X^4$, across the first quadrant, crosses the line $Y = X$ |
| XS | New X, based on points XP, YP and XR, YR, where chord crosses $Y = X$ |
| YP | Starts as zero (i.e., $Y = A - B X^4$ evaluated for $X = \left(\frac{A}{B}\right)^4$) then set from YR |
| YS | Evaluation of $Y = A - B X^4$ for $X = XS$ |
| XNUM | Numerator of new X (i.e., XS) in iteration loop |

## Labeled Common Arrays

None

## Mass Storage File Records Used

None

## Error Messages

The error messages that are provided for negative values A and/or B cannot occur with the simplified form used in SWEEP.

## FUNCTION TTO

### General Description

| | |
|---|---|
| Deck name: | TTO |
| Entry name: | TTO (XMACH, TO, IER) |
| Called by: | TEMPER |
| Subroutines called: | None |

This routine determines the temperature of the air in the vicinity of the moving air vehicle. The basic derivation relates the change in enthalphy of air to the change in kinetic energy. The variation of specific heat with temperature is taken into account.

### Arrays and Variable Used

The input arguments are:

| | |
|---|---|
| XMACH | Free-stream mach number |
| TO | Free-stream temperature - the temperature obtained from function TEMALT |

### Arrays and Variables Calculated

| | |
|---|---|
| TTO | The resulting temperature is returned directly to the using expression |
| IER | Output argument, if zero calculation criteria were met; if one, the solution did not converge, and the temperature returned is based on constant GAMMA. |

### Scratch Variable

| | |
|---|---|
| DELTAT | Incremental temperature |
| EXP1 | Term in specific heat equation |
| EXP2 | Term in specific heat equation |
| GAMMA | Ratio of specific heats, SPHT/SPHTV |
| IND | Iteration counter |

| | |
|---|---|
| SPHT | Specific heat at constant pressure, at TO |
| SPHTA | Adjusted, or new, specific heat |
| SPHTI | Initial, or prior, specific heat |
| SPHTV | Specific heat at constant volume at TO |
| SPSD | Speed of sound |
| VEL | Velocity |

Labeled Common Arrays

None

Mass Storage File Records Used

None

Error Messages

None

SUBROUTINE WHVMAT

General Discussion

| | |
|---|---|
| Deck name: | WHVMAT |
| Entry name: | WHVMAT |
| Called by: | WHVQQ |
| Subroutines called: | None |

Using the material records specified for the major surfaces, this routine sets up tables of compression yielded stress and shear modulus versus temperature. The stress at 80° F is determined by interpolation in the stress table. Figure 25 is a sample of the output from this routine which is printed when IP(41) is zero.

Arrays and Variables Used

Each material record used is read into the TM array; five items are used from each temperature block in the record. Locations are given for the first temperature. For each succeeding temperature, add 25 to the locations.

```
** WHVMAT - IP(A1) **

*** WING ***
7075-T6 AL BARE PLTE 0.25 TO 0.50 IN. MIL-HDBK-5 B DATA PSI.
REF TABLE 3.2.7.0(B) PAGE 334   4-06-72

TEMPERATURE      STRESS (PSI)          G (PSI)

    80.            71000.              3947368.
     0.                0.                   0.
     0.                0.                   0.
     0.                0.                   0.
     0.                0.                   0.
     0.                0.                   0.

STRESS AT 80 DEGREES     71000.

*** HORIZONTAL TAIL ***
2024-T851 AL BARE PLATE 0.5 TO 1.0 IN. REF-AF1.90/1.10
120 HRS AT 290 DEG  MIL-HDBK-5 S DATA  10-24-69

TEMPERATURE      STRESS (PSI)          G (PSI)

    80.            58500.              4022551.
   200.            55500.              3903346.
   300.            49500.              3755523.
     0.                0.                   0.
     0.                0.                   0.
     0.                0.                   0.

STRESS AT 80 DEGREES     58500.

*** VERTICAL TAIL ***
2024-T851 AL BARE PLATE 0.5 TO 1.0 IN. REF-AF1.90/1.10
120 HRS AT 290 DEG  MIL-HDBK-5 S DATA  10-24-69

TEMPERATURE      STRESS (PSI)          G (PSI)

    80.            58500.              4022551.
   200.            55500.              3903346.
   300.            49500.              3755523.
     0.                0.                   0.
     0.                0.                   0.
     0.                0.                   0.

STRESS AT 80 DEGREES     58500.
```

Figure 25. Sample output - stress and shear modulus versus temperature.

| TM Location | Variable Name | Description |
|---|---|---|
| 110 | TEMP | Temperature (° F) for the following data: |
| 111 | | Mu, Poisson's ratio |
| 112 | | $\epsilon_{cPL}$, compression strain at the proportional limit |
| 114 | | $\sigma_{cPL}$, compression stress at the proportional limit |
| 118 | ST | $F_{cy}$, compression stress at yield point |

Locations TM(261) through TM(300) containing the material title are also used if the printout is requested.

## Arrays and Variables Calculated

| | |
|---|---|
| GH | Shear modulus at library temperature for horizontal tail material |
| GV | Shear modulus at library temperature for vertical tail material |
| GW | Shear modulus at library temperature for wing material |
| NHOR | Number of values in horizontal tail stress-G-temperature table |
| NVER | Number of values in vertical tail stress-G-temperature table |
| NWING | Number of values in wing stress-G-temperature table |
| STH | Compression yield stress for horizontal tail material at library temperatures |
| STV | Compression yield stress for vertical tail material at library temperatures |
| STW | Compression yield stress for wing material at library temperatures |
| S80H | Compression yield stress for horizontal tail material at 80° F |
| S80V | Compression yield stress for vertical tail material at 80° F |
| S80W | Compression yield stress for wing material at 80° F |
| TEMPH | Horizontal tail material library temperatures |
| TEMPV | Vertical tail material library temperatures |
| TEMPW | Wing material library temperatures |

## Scratch Arrays and Variables

| | |
|---|---|
| DMNOH | From XMISC(19) material number for horizontal |
| DMNOV | From XMISC(23) material number for vertical |
| DMNOW | From XMISC(15) material number for wing |
| G | Shear modulus, working array |
| I | Counter for which surface in work |
| IEL | Material record number |

| | |
|---|---|
| L1 | First point subscript in interpolation for 80 degrees |
| L2 | Second point subscript in interpolation for 80 degrees |
| N | Scratch counter |
| NN | Scratch counter |
| NTEMP | Number of temperature blocks in material record being used |
| ST | Compression stresses, working array |
| S80 | Stress at 80 degrees, working location |
| TEMP | Temperatures from material record, working array |

Labeled Common Arrays

IP(41) from block IPRINT is the print/no-print indicator. When IP(41) is zero, the stress and G versus temperature tables are printed.

XMISC values, from block MISC, used are the material numbers itemized under Scratch.

Mass Storage File Records Used

Those material records between 41 and 60 whose numbers are specified by material identification number plus 40 are read.

Error Messages

None

SUBROUTINE WHVQQ

General Description

Deck name: WHVQQ
Entry name: WHVQQ
Called by: OLAY3
Subroutines called: WHVMAT, SVFTAB, QINC, QSUB, TEMPER
Function routines used: PRESH, TEMALT

This subroutine controls the search for wing, horizontal tail, and vertical tail, design dynamic pressure, corrected for compressibility effects, and the corresponding material shear modulus. For variable sweep wing vehicles, this routine determines the design dynamic pressure and shear

modulus for the wing in both the forward and aft positions. This routine also scans the limit speed profile for the maximum dynamic pressure for T-tail flutter evaluation. In order to develop these data, the following procedure is used:

1. Call WHVMAT, which enters the material library files and sets up tables of compression yield stress and shear modulus for the wing, horizontal tail, and vertical tail structural materials. These data are stored in the blank common array SVF.

2. Read record 38, speed-altitude profile data and flutter margin.

3. Call SVFTAB, which interpolates the flutter parameter tables from record 12 to obtain flutter parameter versus mach number for each of the surfaces.

4. Use function routines PRESH and TEMALT and call QINC to calculate dynamic pressure at the limit speed-altitude profile (wings fixed or aft) points.

5. Set up a table of mach numbers and dynamic pressures representing the design flutter margin boundary at constant altitude.

6. Call QSUB to calculate the equivalent dynamic pressure for incompressible torsional divergence corresponding to each point on the flutter mach-altitude profile.

7. Scan the data from QSUB for the maximum design dynamic pressure and the corresponding mach number for the wing.

| NOTE: PROGRAM LIMITATION |
|---|
| This pressure is a maximum, though not in general the critical, value for lifting surface flutter evaluations. Correct values should be determined through an evaluation of the ratio of dynamic pressure to shear modulus. Such an approach would account for the influence of temperature upon the critical flutter condition. However, the values of shear modulus for most aircraft structural materials does not change significantly in the transonic mach number ranges where flutter is usually critical. Cases in which the value of shear modulus, G, does vary significantly in the mach number range where flutter is critical could be |

conveniently handled by first computing the variation of G along the $V_L$ profile (defined as $G_M$) and then multiplying the flutter boundary determined by step 3 by $\sqrt{G_{ref}/G_M}$ ; where $G_{ref}$ is the shear modulus at room temperature. The procedure of step 7 could then use this modified flutter boundary, and the maximum design dynamic pressure obtained would be the true flutter critical value

8. Repeat steps 6 and 7 for the horizontal and vertical tails.

9. Set up a table of mach numbers and dynamic pressures representing the design flutter margin boundary at constant mach number.

10. Repeat steps 6 through 8 for constant mach number flutter margin.

11. If aircraft has a variable-sweep wing, do steps 4, 5, 6, 7, and 9 for limit speed-altitude profile with wing in forward position.

12. Calculate T-tail flutter parameter at each speed-altitude profile point for flutter margin at constant altitude and constant mach number.

13. Scan T-tail flutter parameter and dynamic pressure data for design dynamic pressure.

14. Call TEMPER to calculate structure temperature at design flutter point for:

    a. Wing fixed or aft
    b. Horizontal tail
    c. Vertical tail
    d. Wing forward (variable sweep only)
    e. T-tail flutter

15. Interpolate material property data for shear modulus at structure temperature associated with design flutter points.

16. Write flutter design data in record 38 and labeled common block MISC for use in the wing and empennage weight estimation module.

## Arrays and Variable Used

Common region variables:

| | |
|---|---|
| GH | Shear modulus at library temperatures for horizontal tail material |
| GV | Shear modulus at library temperatures for vertical tail material |
| GW | Shear modulus at library temperatures for wing material |
| NHOR | Number of values in horizontal tail stress-G-temperature tables |
| NVER | Number of values in vertical tail stress-G-temperature tables |
| NWING | Number of values in wing stress-G-temperature tables |
| QQ | Dynamic pressure corrected for compressibility effect at speed profile altitude |
| TEMPH | Horizontal tail material library temperatures; NHOR values |
| TEMPV | Vertical tail material library temperatures; NVER values |
| TEMPW | Wing material library temperatures; NWING values |
| TFH | Flutter parameter array at table mach numbers for horizontal tail |
| TFV | Flutter parameter array at table mach numbers for vertical tail |
| TFW | Flutter parameter array at table mach numbers for wing fixed or aft |
| TFWF | Flutter parameter array at table mach numbers for wing forward |
| VFF | Flutter speed margin |
| XMATOP | Mach number associated with QQ |

Program region variables (refer to Table 17):

ALTA, ALTF, FM, XMA, XMF

Program region variables (refer to Table 18):

CTTD1, CTTD2, CTT1, CTT2, PC

## Arrays and Variables Calculated

Common region variables:

| | |
|---|---|
| Q | Incompressible dynamic pressure at design mach numbers |
| TF | Flutter parameter array for the surface being evaluated |
| XMN1 | Design mach numbers |

TABLE 17. SPAL ARRAY VARIABLES

| Loc | Name Variable | Description |
|---|---|---|
| 1 | | Weight of horizontal tail per side |
| 2 | | $Y_{cg}$, butt plane of horizontal tail cg, per side |
| 3 | | $X_{cg}$, fuselage station of horizontal tail cg, per side |
| 4 | | $Z_{cg}$, water plane of horizontal tail cg, per side |
| 5 | | $I_{pitch}$, pitch inertia about $X_{cg}$ per vehicle |
| 6 | | $I_{roll}$ |
| 7 | | $I_{yaw}$, yaw inertia about $X_{cg}$ per vehicle |
| 8 | XMTD | $M_{TT}$, design mach number for T-tail |
| 9 | QTT | $Q_{TT}$, design dynamic pressure for T-tail |
| 10 | | $G_{TT}$, shear modulus for T-tail |
| 11 | CTT | $K_{TT}$, K-factor for T-tail |
| 12 | | $\Gamma_{H}$, dihedral angle of horizontal tail |
| 13 | | Not used |
| 14 | | Not used |
| 15 | | Not used |
| 16 | MF | Flutter speed margin |
| 17 | ALTA(1) | Altitude 1 of $M_L$ diagram<br>to |
| . | . | |
| 25 | ALTA(9) | Altitude 9 of $M_L$ diagram |
| 26 | XMA(1) | Mach number 1 of $M_L$ diagram<br>to |
| . | . | |
| 34 | XMA(9) | Mach number 9 of $M_L$ diagram |
| 35 | ALTF(1) | Altitude 1 of $M_L$ diagram for wing in forward position<br>to |
| . | . | |
| 37 | ALTF(3) | Altitude 3 of $M_L$ diagram for wing in forward position |
| 38 | XMF(1) | Mach number 1 of $M_L$ diagram for wing in forward position<br>to |
| . | . | |
| 40 | XMF(3) | Mach number 3 of $M_L$ diagram for wing in forward position |
| 41 | AWF | Critical flutter altitude for wing in forward position |
| 42 | XMWF | Mach number corresponding to AWF |
| 43 | TWF | Temperature of skin at AWF, XMWF |
| 44 | QQWF | Dynamic pressure corrected for compressibility at AWF, XMWF |
| 45 | GWF | Shear modulus at TWF |
| 46 | TSF(1) | Wing temperature for flutter design for wing fixed or - aft - ° F |
| 47 | TSF(2) | Horizontal tail temperature for flutter design - ° F |
| 48 | TSF(3) | Vertical tail temperature for flutter design - ° F |
| 49 | TFTT | Vertical tail temperature for T-tail flutter design - ° F |
| 50 | | Not used |

TABLE 18. GJDAT ARRAY VARIABLES

| Loc | Value | Variable Name | Description |
|---|---|---|---|
| 1 | | | Not used |
| . | | | |
| 17 | | | Not used |
| 18 | 20.0 | PC | Number of flutter parameter points |
| 19 | 0.0 | CTTD1 | Dihedral angle for flutter parameter values in CTT1 array, deg |
| 20 | 15.0 | CTTD2 | Dihedral angle for flutter parameter values in CTT2 array, deg |
| 21 | 0.0 | CTTM(1) | Mach number at point 1 |
| 22 | 0.82 | | |
| 23 | 0.85 | | |
| 24 | 0.88 | | |
| 25 | 0.92 | | |
| 26 | 0.95 | | |
| 27 | 0.97 | | |
| 28 | 1.0 | | |
| 29 | 1.04 | to | |
| 30 | 1.07 | | |
| 31 | 1.12 | | |
| 32 | 1.2 | | |
| 33 | 1.3 | | |
| 34 | 1.4 | | |
| 35 | 1.5 | | |
| 36 | 1.6 | | |
| 37 | 1.7 | | |
| 38 | 1.8 | | |
| 39 | 1.9 | | |
| 40 | 2.0 | CTTM(20) | Mach number at point 20 |
| 41 | 146000.0 | CTT1(1) | T-tail flutter parameter for CTTD1 at CTTM(1) |
| 42 | 146000.0 | | |
| 43 | 146000.0 | | |
| 44 | 149000.0 | | |
| 45 | 154000.0 | | |
| 46 | 178000.0 | | |
| 47 | 186000.0 | | |
| 48 | 190000.0 | | |
| 50 | 188000.0 | | |
| 49 | 191000.0 | | |
| 50 | 188000.0 | | |

TABLE 18. GJDAT ARRAY VARIABLES (CONCL)

| Loc | Value | Variable Name | Description |
|---|---|---|---|
| 51 | 184000.0 | | |
| 52 | 172000.0 | | |
| 53 | 158000.0 | | |
| 54 | 142000.0 | | |
| 55 | 130000.0 | | |
| 56 | 121000.0 | | |
| 57 | 113000.0 | | |
| 58 | 108000.0 | | |
| 59 | 105000.0 | | |
| 60 | 102000.0 | CTT1(20) | T-tail flutter parameter for CTTD1 at CTTM(20) |
| 61 | 344000.0 | CTT2(1) | T-tail flutter parameter for CTTD2 at CTTM(1) |
| 62 | 355000.0 | | |
| 63 | 355000.0 | | |
| 64 | 355000.0 | | |
| 65 | 355000.0 | | |
| 66 | 353000.0 | | |
| 67 | 351000.0 | | |
| 68 | 350000.0 | | |
| 69 | 345000.0 | | |
| 70 | 340000.0 | | |
| 71 | 340000.0 | | |
| 71 | 331000.0 | | |
| 72 | 310000.0 | | |
| 73 | 284000.0 | | |
| 74 | 264000.0 | | |
| 75 | 255000.0 | | |
| 76 | 249000.0 | | |
| 77 | 244000.0 | | |
| 78 | 240000.0 | | |
| 79 | 238000.0 | | |
| 80 | 235000.0 | CTT2(20) | T-tail flutter parameter for CTTD2 at CTTM(20) |
| 81 | | | Not used |
| . | | | |
| 100 | | | Not used |

Program region variables (refer to Table 17):

AWF, CTT, GWF, QQWF, QTT, TFTT, TSF, TWF, XMAF, SMTD, and SPAL(10)

N Number of speed-altitude profile points

## Scratch Arrays and Variables

| | |
|---|---|
| AL | Altitude at design flutter for wing fixed or aft, horizontal tail, and vertical tail |
| DTT | Absolute value of horizontal tail dihedral |
| FACT | T-tail flutter parameter times dynamic pressure at speed-altitude profile points |
| FACT1 | Maximum of values in FACT array |
| FMM | Flutter speed margin squared; margin on dynamic pressure |
| G | Shear modulus, working array |
| GG | Shear modulus at structure temperature |
| GHT | Horizontal tail structural material shear modulus at design flutter point |
| GTEMP | Structure temperature, working array |
| GVT | Vertical tail structural material shear modulus at design flutter point |
| GWING | Wing structural material shear modulus at design flutter point with wings fixed or aft |
| I | Scratch counter |
| IC | Curve interpolation point counter |
| ITT | Speed-altitude profile point at design T-tail flutter |
| IFEXT | Indicator to designate extrapolation of T-tail flutter parameter data |
| IER | Error indicator |
| IT | Index of point on mach-altitude envelope corresponding to maximum design flutter dynamic pressure for wing fixed or aft, horizontal tail, and vertical tail |
| ITW | Speed-altitude profile point at design flutter for wings forward |
| J | Counter for search<br>1 = test on flutter speed margin<br>2 = test of flutter dynamic pressure margin |
| K | Scratch counter |
| L1 | Interpolation lower limit point indicator |
| L2 | Interpolation upper limit point indicator |
| NP | Number of T-tail flutter parameter points |
| NTEMP | Number of values in stress-G-temperature tables, working array |
| PO | Local static pressure, $lb/ft^2$ |
| PRE | Local static pressure, $lb/in.^2$ |
| QA | Incompressible dynamic pressure at limit speed-altitude profile points |

| | |
|---|---|
| SFX | Sun flux, BTU/hr/ft$^2$ |
| TEMP | Structure temperature, working array |
| TLOC | Local static temperature, $^\circ$ R |
| TO | Local static temperature, $^\circ$ R |
| TSKR | Equilibrium skin temperature, $^\circ$ R |
| TT | T-tail flutter parameter at speed-altitude profile points |
| TTOT | Total temperature, $^\circ$ R |
| XM | Mach number at design flutter for wing fixed or aft, horizontal tail, and vertical tail. |
| XMTT | Mach number at limit speed-altitude profile T-tail flutter design point |

Labeled Common Arrays

| | |
|---|---|
| IP(41) | Print/no-print indicator<br>0 = print dynamic pressure and design shear modulus data (figures 26 through 29)<br>1 = no print |
| XMISC(5) | Dynamic pressure for wing design, lb/ft$^2$ |
| XMISC(6) | Dynamic pressure for horizontal tail design, lb/ft$^2$ |
| XMISC(7) | Dynamic pressure for vertical tail design, lb/ft$^2$ |
| XMISC(15) | Wing structural material identification number; established in input data processing module |
| XMISC(19) | Horizontal tail structural material identification number; established in input data processing module |
| XMISC(23) | Vertical tail structural material identification number; established in input data processing module |
| XMISC(28) | Wing structural material shear modulus at design flutter point, lb/in.$^2$ |
| XMISC(29) | Horizontal tail structural material shear modulus at design flutter point, lb/in.$^2$ |
| XMISC(30) | Vertical tail structural material shear modulus at design flutter point, lb/in.$^2$ |
| XMISC(53) | Vertical tail-type indicator<br>- 1 = single tail<br>0 = dual tail<br>1 = T-tail |

Mass Storage File Records Used

| | |
|---|---|
| Record 37 | GJDAT array, T-tail flutter parameter tables |
| Record 38 | SPAL array, speed-altitude profile and T-tail inertia data, is read and written |

FLUTTER SPEED MARGIN = 1.20 ** WHV0J - IP(41) **

WING FIXED OR AFT

| SPEED-ALTITUDE PROFILE POINTS | | | FLUTTER DESIGN | | |
|---|---|---|---|---|---|
| ALTITUDE FEET | MACH NUMBER | DYNAMIC PRESSURE | MACH NUMBER | DYNAMIC PRESSURE | COMPRESSIBLE DYNAMIC PRESSURE |
| 0. | 1.2800 | 2426.3 | 1.5360 | 3493.9 | 1657.9 |
| 5000. | 1.4419 | 2562.1 | 1.7303 | 3689.5 | 1379.6 |
| 10000. | 1.6275 | 2697.9 | 1.9530 | 3885.0 | 1140.3 |
| 15000. | 1.7984 | 2703.6 | 2.1581 | 3893.1 | 935.8 |
| 20000. | 1.9950 | 2709.2 | 2.3940 | 3901.2 | 762.1 |
| 27500. | 2.1946 | 2371.0 | 2.6335 | 3414.3 | 551.2 |
| 35000. | 2.4150 | 2032.9 | 2.8980 | 2927.3 | 390.2 |
| 37500. | 2.5185 | 1961.1 | 3.0222 | 2823.9 | 346.1 |
| 40000. | 2.6250 | 1889.2 | 3.1500 | 2720.5 | 312.7 |

HORIZONTAL TAIL

| SPEED-ALTITUDE PROFILE POINTS | | | FLUTTER DESIGN | | |
|---|---|---|---|---|---|
| ALTITUDE FEET | MACH NUMBER | DYNAMIC PRESSURE | MACH NUMBER | DYNAMIC PRESSURE | COMPRESSIBLE DYNAMIC PRESSURE |
| 0. | 1.2800 | 2426.3 | 1.5360 | 3493.9 | 1589.4 |
| 5000. | 1.4419 | 2562.1 | 1.7303 | 3689.5 | 1322.6 |
| 10000. | 1.6275 | 2697.9 | 1.9530 | 3885.0 | 1093.2 |
| 15000. | 1.7984 | 2703.6 | 2.1581 | 3893.1 | 897.1 |
| 20000. | 1.9950 | 2709.2 | 2.3940 | 3901.2 | 730.6 |
| 27500. | 2.1946 | 2371.0 | 2.6335 | 3414.3 | 528.4 |
| 35000. | 2.4150 | 2032.9 | 2.8980 | 2927.3 | 398.2 |
| 37500. | 2.5185 | 1961.1 | 3.0222 | 2823.9 | 367.1 |
| 40000. | 2.6250 | 1889.2 | 3.1500 | 2720.5 | 339.7 |

VERTICAL TAIL

| SPEED-ALTITUDE PROFILE POINTS | | | FLUTTER DESIGN | | |
|---|---|---|---|---|---|
| ALTITUDE FEET | MACH NUMBER | DYNAMIC PRESSURE | MACH NUMBER | DYNAMIC PRESSURE | COMPRESSIBLE DYNAMIC PRESSURE |
| 0. | 1.2800 | 2426.3 | 1.5360 | 3493.9 | 1551.7 |
| 5000. | 1.4419 | 2562.1 | 1.7303 | 3689.5 | 1291.2 |
| 10000. | 1.6275 | 2697.9 | 1.9530 | 3885.0 | 1067.3 |
| 15000. | 1.7984 | 2703.6 | 2.1581 | 3893.1 | 875.8 |
| 20000. | 1.9950 | 2709.2 | 2.3940 | 3901.2 | 713.2 |
| 27500. | 2.1946 | 2371.0 | 2.6335 | 3414.3 | 528.0 |
| 35000. | 2.4150 | 2032.9 | 2.8980 | 2927.3 | 399.4 |
| 37500. | 2.5185 | 1961.1 | 3.0222 | 2823.9 | 365.4 |
| 40000. | 2.6250 | 1889.2 | 3.1500 | 2720.5 | 335.5 |

Figure 26. Sample output - dynamic pressure to satisfy flutter speed margin.

FLUTTER % MARGIN = 1.4400 ** AHVOL - IP(41) **

WING FIXED OR AFT

| SPEED-ALTITUDE PROFILE POINTS | | | FLUTTER DESIGN | | |
|---|---|---|---|---|---|
| ALTITUDE FEET | MACH NUMBER | DYNAMIC PRESSURE | MACH NUMBER | DYNAMIC PRESSURE | COMPRESSIBLE DYNAMIC PRESSURE |
| 0. | 1.2800 | 2426.3 | 1.2800 | 3493.9 | 2387.4 |
| 5000. | 1.4419 | 2562.1 | 1.4419 | 3689.5 | 1986.6 |
| 10000. | 1.6275 | 2697.9 | 1.6275 | 3885.0 | 1642.1 |
| 15000. | 1.7984 | 2703.6 | 1.7984 | 3893.1 | 1347.6 |
| 20000. | 1.9950 | 2709.2 | 1.9950 | 3901.2 | 1097.4 |
| 27500. | 2.1946 | 2371.0 | 2.1946 | 3414.3 | 793.7 |
| 35000. | 2.4150 | 2032.9 | 2.4150 | 2927.3 | 561.9 |
| 37500. | 2.5185 | 1961.1 | 2.5185 | 2823.9 | 498.4 |
| 40000. | 2.6250 | 1889.2 | 2.6250 | 2720.5 | 442.0 |

HORIZONTAL TAIL

| SPEED-ALTITUDE PROFILE POINTS | | | FLUTTER DESIGN | | |
|---|---|---|---|---|---|
| ALTITUDE FEET | MACH NUMBER | DYNAMIC PRESSURE | MACH NUMBER | DYNAMIC PRESSURE | COMPRESSIBLE DYNAMIC PRESSURE |
| 0. | 1.2800 | 2426.3 | 1.2800 | 3493.9 | 2286.7 |
| 5000. | 1.4419 | 2562.1 | 1.4419 | 3689.5 | 1904.5 |
| 10000. | 1.6275 | 2697.9 | 1.6275 | 3885.0 | 1574.2 |
| 15000. | 1.7984 | 2703.6 | 1.7984 | 3893.1 | 1291.9 |
| 20000. | 1.9950 | 2709.2 | 1.9950 | 3901.2 | 1052.0 |
| 27500. | 2.1946 | 2371.0 | 2.1946 | 3414.3 | 760.9 |
| 35000. | 2.4150 | 2032.9 | 2.4150 | 2927.3 | 538.7 |
| 37500. | 2.5185 | 1961.1 | 2.5185 | 2823.9 | 477.8 |
| 40000. | 2.6250 | 1889.2 | 2.6250 | 2720.5 | 423.7 |

VERTICAL TAIL

| SPEED-ALTITUDE PROFILE POINTS | | | FLUTTER DESIGN | | |
|---|---|---|---|---|---|
| ALTITUDE FEET | MACH NUMBER | DYNAMIC PRESSURE | MACH NUMBER | DYNAMIC PRESSURE | COMPRESSIBLE DYNAMIC PRESSURE |
| 0. | 1.2800 | 2426.3 | 1.2800 | 3493.9 | 2234.4 |
| 5000. | 1.4419 | 2562.1 | 1.4419 | 3689.5 | 1859.3 |
| 10000. | 1.6275 | 2697.9 | 1.6275 | 3885.0 | 1536.8 |
| 15000. | 1.7984 | 2703.6 | 1.7984 | 3893.1 | 1261.2 |
| 20000. | 1.9950 | 2709.2 | 1.9950 | 3901.2 | 1027.0 |
| 27500. | 2.1946 | 2371.0 | 2.1946 | 3414.3 | 742.8 |
| 35000. | 2.4150 | 2032.9 | 2.4150 | 2927.3 | 525.9 |
| 37500. | 2.5185 | 1961.1 | 2.5185 | 2823.9 | 466.5 |
| 40000. | 2.6250 | 1889.2 | 2.6250 | 2720.5 | 422.5 |

Figure 27. Sample output - dynamic pressure to satisfy flutter dynamic pressure margin.

** WHVQQ - IP(4j) **

T-TAIL FLUTTER DIHEDRAL = 0.0

FLUTTER SPEED MARGIN = 1.15

| SPEED-ALTITUDE PROFILE POINTS ALTITUDE FEET | MACH NUMBER | DYNAMIC PRESSURE | FLUTTER DESIGN MACH NUMBER | FLUTTER DESIGN DYNAMIC PRESSURE | CIT | Q*CIT |
|---|---|---|---|---|---|---|
| 0. | .6000 | 533.1 | .6900 | 705.1 | .146000E+06 | .102938E+09 |
| 5000. | .6499 | 520.4 | .7473 | 688.2 | .146000E+06 | .100484E+09 |
| 10000. | .7060 | 507.7 | .8119 | 671.4 | .146000E+06 | .980275E+08 |
| 15000. | .7688 | 494.0 | .8841 | 653.3 | .156439E+06 | .102204E+09 |
| 20000. | .8400 | 480.3 | .9660 | 635.2 | .189200E+06 | .120178E+09 |
| 21250. | .8548 | 471.9 | .9830 | 624.1 | .190435E+06 | .118847E+09 |
| 22500. | .8700 | 463.5 | 1.0005 | 613.0 | .190962E+06 | .117055E+09 |
| 36250. | .8700 | 248.5 | 1.0005 | 328.7 | .190962E+06 | .627603E+08 |
| 50000. | .8700 | 128.3 | 1.0005 | 169.7 | .190962E+06 | .324092E+08 |

FLUTTER Q MARGIN = 1.3225

| SPEED-ALTITUDE PROFILE POINTS ALTITUDE FEET | MACH NUMBER | DYNAMIC PRESSURE | FLUTTER DESIGN MACH NUMBER | FLUTTER DESIGN DYNAMIC PRESSURE | CIT | Q*CIT |
|---|---|---|---|---|---|---|
| 0. | .6000 | 533.1 | .6000 | 705.1 | .146000E+06 | .102938E+09 |
| 5000. | .6499 | 520.4 | .6499 | 688.2 | .146000E+06 | .100484E+09 |
| 10000. | .7060 | 507.7 | .7060 | 671.4 | .146000E+06 | .980275E+08 |
| 15000. | .7688 | 494.0 | .7688 | 653.3 | .146000E+06 | .953834E+08 |
| 20000. | .8400 | 480.3 | .8400 | 635.2 | .148000E+06 | .940084E+08 |
| 21250. | .8548 | 471.9 | .8548 | 624.1 | .149804E+06 | .934898E+08 |
| 22500. | .8700 | 463.5 | .8700 | 613.0 | .152333E+06 | .933760E+08 |
| 36250. | .8700 | 248.5 | .8700 | 328.7 | .152333E+06 | .500647E+08 |
| 50000. | .8700 | 128.3 | .8700 | 169.7 | .152333E+06 | .258532E+08 |

DESIGN POINT

| SPEED-ALTITUDE PROFILE POINTS ALTITUDE FEET | MACH NUMBER | DYNAMIC PRESSURE | FLUTTER DESIGN MACH NUMBER | FLUTTER DESIGN DYNAMIC PRESSURE | CIT | Q*CIT |
|---|---|---|---|---|---|---|
| 20000. | .8400 | 480.3 | .9660 | 635.2 | .189200E+06 | .120178E+09 |

Figure 28. Flutter parameter and dynamic pressure to satisfy T-Tail flutter.

** ANAL.. - IP(41) **

WING FORWARD

FLUTTER SPEED MARGIN = 1.20

| SPEED-ALTITUDE PROFILE POINTS | | | FLUTTER DESIGN | | |
|---|---|---|---|---|---|
| ALTITUDE FEET | MACH NUMBER | DYNAMIC PRESSURE | MACH NUMBER | DYNAMIC PRESSURE | COMPRESSIBLE DYNAMIC PRESSURE |
| 0. | 0.8000 | 947.8 | 0.9600 | 1364.8 | 1363.8 |
| 10000. | 0.9000 | 825.0 | 1.0800 | 1188.1 | 1378.5 |
| 30000. | 1.0000 | 439.9 | 1.2000 | 633.4 | 472.9 |

FLUTTER Q MARGIN = 1.4400

| SPEED-ALTITUDE PROFILE POINTS | | | FLUTTER DESIGN | | |
|---|---|---|---|---|---|
| ALTITUDE FEET | MACH NUMBER | DYNAMIC PRESSURE | MACH NUMBER | DYNAMIC PRESSURE | COMPRESSIBLE DYNAMIC PRESSURE |
| 0. | 0.8000 | 947.8 | 0.8000 | 1364.8 | 1364.8 |
| 10000. | 0.9000 | 825.0 | 0.9000 | 1188.1 | 1201.6 |
| 30000. | 1.0000 | 439.9 | 1.0000 | 633.4 | 671.4 |

*** DESIGN TEMPERATURE, PRESSURE AND G ***

| | PROFILE POINT | ALTITUDE | MACH NO. | TEMPERATURE | PRESSURE | G (PSI) |
|---|---|---|---|---|---|---|
| WING | 1 | 0. | 1.0500 | 161.3 | 2387.4 | 3947369. |
| HORIZONTAL | 1 | 0. | 1.1000 | 170.9 | 2286.7 | 3932267. |
| VERTICAL | 1 | 0. | 1.1250 | 175.9 | 2234.4 | 3927333. |

Figure 29. Sample output - dynamic pressure data for wings forward and surface flutter design data.

Error Messages

*****EXTRAPOLATED ON T-TAIL STIFFNESS COEFF. FOR MACH NO. x.x, CTT = y.y when x.x is mach number and y.y is value of coefficient to be used.

## Section IV

## MODULE FLOW CHARTS AND FORTRAN LISTS

### FLOW CHART USAGE

The automatically generated computer program flow charts (AUTOFLOW) presented in this document include a table of contents, flow charts, and FORTRAN lists of all routines in the module. The 80-column card lists are sequenced and grouped by routine.

Because the AUTOFLOW system used is IBM-oriented, the functions of the BUFFERIN and BUFFEROUT statements are not recognized, but these statements appear in proper order in note boxes. Also, the PROGRAM name does not appear on the main program, and library routines READMS and WRITMS are listed as undefined external references.

### CROSS-REFERENCE LIST

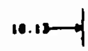
The AUTOFLOW table of contents which precedes the flow charts and FORTRAN lists serves to cross reference the latter two. This table lists the following from left to right:

- The card identification from columns 73 through 80 of this card, or card sequence number. When sequence number is used in place of card identification, it is enclosed in parentheses.
- The page and box number where this card is displayed in a flow chart.
- The FORTRAN statement number from columns 1 through 5 of this card.
- The card identification(s) or sequence number(s) of the card(s) referring to this card (repeated as required).
- The pages and box numbers where the cards referring to this card are displayed in a flow chart (repeated as required).

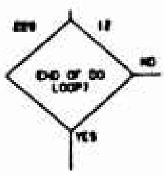
### FLOW CHARTS

The flow charts produced by AUTOFLOW use USASI conventional symbols. Since the flow charts are mechanically drawn from the program source deck, there are no omissions or vague generalizations about the processing within the boxes.

Every box on each page is uniquely numbered and may be referred to from elsewhere in the program. The source of a reference to a box will be indicated by showing the page and box number. If the number is followed by an asterisk, there are multiple references to this point, and the others may be found by using the cross-reference list.

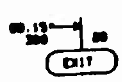


The most-often-used symbol is the decision box. Like all boxes, its box number is above and to the right of the box. Its FORTRAN statement number is above and to the left of the box. The decision choices for the paths are printed.

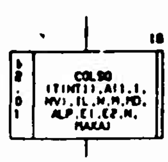


The unconditional transfer connecter has its page number destination printed above or to the left of the box number destination within the connecter. If there is a FORTRAN statement number at the destination, it is printed below the connecter.

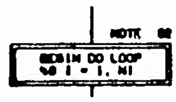
The exit box example shows a connecter from page 9, box 15.



The subroutine call box includes the calling sequence. The page and box numbers of the flow chart of the called subroutine are shown on the left-hand side of the box. The page number is above the box number.



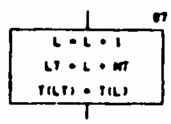
The note box encloses comments of a functional nature,



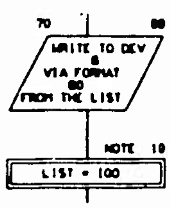

as differentiated from the 21 column comments, which are left justified without a box, that show the comment cards included in the FORTRAN deck.

```
-----100-1
CALLING PROGRAM
A(MA,K)   B(MB,K)
C(MC,K)
OPERATION
C(NI,NJ)=A(NI,NK)
*B(NK,NJ)
-----100-2
CALLING PROGRAM
A(MA,K)   B(MB,K)
C(MC,K)
OPERATION
C(NI,NJ)=(A(NK,NI))
-TRANSPOSE()*B(NK,NJ)
```

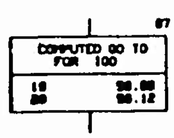
The process box is used to enclose FORTRAN arithmetic statements.




Input and output are shown as communicating with a device. The list used follows, if appropriate:




The computed GØ TØ becomes a branch table showing the page and box number of each of the ordered branches.




The column connecters and initial connecters are the only boxes without external box numbers. The function of the initial connecter is always clear,

but the label given is the symbol in the next FORTRAN card, which is often blank.

The column connecter identifies the page and box number to which it connects.

TABLE OF CONTENTS

FOR

AUTOFLOW CHART SET

```
11/07/73       TABLE OF CONTENTS AND REFERENCES       AUTOFLOW CHART SET - SWEEP                                    PAGE   1
CARD ID   PAGE/BOX    NAME                          REFERENCES  (SOURCE SEQUENCE NO. AND PAGE/BOX)


FORTRAN MODULE     FLUTTER AND TEMPERATURE


CHART TITLE - INTRODUCTORY COMMENTS


CHART TITLE - PROCEDURES

(000017)    2.02  10
(000017)    2.02              (000017)   2.03
(000020)    2.05              (000022)   2.07
(000022)    2.06  11
(000030)    2.11              (000091)   3.05
(000035)    2.13  72          (000032)   2.12     (000032)   2.12     (000032)   2.12
(000039)    2.14  73          (000032)   2.12
(000043)    2.15  74          (000032)   2.12     (000032)   2.12     (000032)   2.12
(000047)    2.16  75          (000032)   2.12
(000051)    2.17  76          (000032)   2.12     (000032)   2.12     (000032)   2.12     (000032)   2.12
(000055)    2.18  77          (000032)   2.12     (000032)   2.12     (000032)   2.12     (000032)   2.12
(000059)    2.19  78          (000032)   2.12
(000060)    2.20  81          (000064)   2.21
(000063)    2.21  84          (000032)   2.12
(000068)    2.22  85          (000032)   2.12     (000032)   2.12     (000032)   2.12
(000070)    2.23  86          (000032)   2.12     (000032)   2.12
(000073)    2.24  87          (000037)   2.13     (000041)   2.14     (000045)   2.15     (000049)   2.16     (000053)   2.17
                              (000057)   2.18     (000061)   2.20     (000068)   2.22
(000081)    3.01  98          (000076)   2.25
(000086)    3.03  701         (000084)   3.02
(000091)    3.05  123         (000076)   2.25     (000084)   3.02

CHART TITLE - NON-PROCEDURAL STATEMEN S


CHART TITLE - INTRODUCTORY COMMENTS


CHART TITLE - FUNCTION  HBL(XMACH,P0,T0,TSK,DIST,XLAMDA,REY,METHOD,IER,TOTI,TAW)

(000171)    7.02  10
(000172)    7.03  11
(000176)    7.04  13          (000170)   7.01
(000177)    7.05  14
(000178)    7.06  15          (000173)   7.03     (000175)   7.07     (000182)   7.09
(000174)    7.07  12          (000171)   7.02
(000180)    7.08  20          (000176)   7.04
(000181)    7.09  21
(000183)    8.01  22          (000180)   7.08
(000184     8.02  23
(000186)    8.03  24          (000183)   8.01
(000189)    8.05  25
(000190)    8.08  26          (000188)   8.04
(000206)    8.13  100         (000205)   8.12     (000205)   8.12
(000207)    8.14  30
(000208)    8.15  35          (000206)   8.13
(000210)    8.17  40
(000215)    8.20  50
(000217)    8.21  60          (000209)   8.16
(000225)    8.25  65
(000226)    8.26  70          (000224)   8.24     (000231)   8.29
(000229)    8.28  80
(000232)    8.30  90
(000233)    8.31  110         (000228)   8.27
(000235)    9.01  300         (000205)   8.12     (000205)   8.12
(000239)    9.03  130
(000241)    9.05  140
(000242)    9.06  150         (000238)   9.02     (000240)   9.04
(000244)    9.08  180
(000245)    9.09  170         (000243)   9.07
(000252)    9.13  180
(000255)    9.15  190
(000256)    9.16  210         (000254)   9.14
(000258)    9.17  220         (000251)   9.12
(000268)    9.22  229
(000269)    9.23  230         (000257)   9.16
```

```
CARD ID   PAGE/BOX   NAME                 REFERENCES (SOURCE SEQUENCE NO. AND PAGE/BOX)

(000274)   9.26  1000         (000179)   7.08    (000214)   8.19    (000218)   8.20    (000234)   8.31

CHART TITLE - INTRODUCTORY COMMENTS


CHART TITLE - FUNCTION  PRESH(ALT)

(000287)   11 03  10
(000289)   11 04  20          (000286)  11.02
(000290)   11.05  30          (000286)  11.02
(000293)   11 06  40          (000289)  11.04
(000294)   11 07  50
(000297)   11 08  60          (000293)  11.06
(000299)   11.10  65
(000301)   11 11  100         (000288)  11.03    (000292)  11.05    (000296)  11.07    (000298)  11.09

CHART TITLE - NON-PROCEDURAL STATEMENTS


CHART TITLE - INTRODUCTORY COMMENTS


CHART TITLE - SUBROUTINE  QINC(XM,PO,TO,Q)

(000310)   14 01  QINC        (001331)  59.17-X  (001415)  61.10-X

CHART TITLE - INTRODUCTORY COMMENTS


CHART TITLE - SUBROUTINE  OSUB(L,IM)

(000351)   16 01  OSUB        (001349)  59.26-X  (001424)  61.18-X
(000353)   16 02              (000389)  17.15
(000357)   16 04  47          (000355)  16.03
(000361)   17.01  48          (000355)  16.03
(000362)   17 02              (000368)  17.05
(000363)   17 03  49
(000364)   17.04  52          (000362)  17.02
(000368)   17 05  50          (000363)  17.03
(000370)   17 06  54          (000367)  17.04
(000373)   17.08              (000376)  17.10
(000374)   17 09  62
(000378)   17 10  60          (000373)  17.08
(000380)   17.12  71          (000378)  17.11
(000384)   17 13  72          (000378)  17.11
(000387)   17.14  69          (000359)  16.04    (000382)  17.12
(000389)   17 15  100

CHART TITLE - NON-PROCEDURAL STATEMENTS


CHART TITLE - INTRODUCTORY COMMENTS


CHART TITLE - FUNCTION  SOLARG(ALT)

(000405)   20 03  10
(000407)   20.04  20          (000404)  20.02
(000408)   20.05  30          (000406)  20.03

CHART TITLE - INTRODUCTORY COMMENTS


CHART TITLE - SUBROUTINE  SVFTAB

(000417)   22.01  SVFTAB      (001305)  59.03-X
(000444)   22.02              (000447)  22.04
(000447)   22.03  699
(000465)   22.09              (000581)  25.04
(000467)   22.11  301         (000465)  22.09
(000469)   22.12  305         (000467)  22.11
(000475)   22.13  302         (000465)  22.09
(000478)   22.14  332
(000481)   22.15  303         (000465)  22.09
(000483)   22.16  306         (000481)  22.15
(000500)   23.01  204         (000495)  23.08
(000502)   23.03  205
(000502)   23.03              (000502)  23.04
```

| CARD ID | PAGE/BOX | NAME | REFERENCES (SOURCE SEQUENCE NO. AND PAGE/BOX) |
|---|---|---|---|
| (000488) | 23 05 304 | | (000485) 22 09 |
| (000490) | 23.06 307 | | (000488) 23 05 |
| (000494) | 23.07 308 | | (000472) 22 12 (000479) 22.14 (000486) 22.18 |
| (000495) | 23.08 200 | | |
| (000496) | 23.09 201 | | |
| (000498) | 23.11 202 | | |
| (000498) | 23.11 | | (000498) 23.12 |
| (000504) | 23.13 210 | | (000494) 23.07 |
| (000506) | 23.15 215 | | |
| (000506) | 23.15 | | (000506) 23.16 |
| (000508) | 23.17 220 | | (000503) 23.04 (000499) 23.12 |
| (000509) | 23.18 230 | | |
| (000511) | 23.20 235 | | |
| (000511) | 23.20 | | (000511) 23.21 |
| (000513) | 23.22 240 | | (000508) 23.17 |
| (000515) | 23 24 245 | | |
| (000515) | 23.24 | | (000515) 23.25 |
| (000517) | 23.26 250 | | (000512) 23.21 |
| (000518) | 23.27 260 | | |
| (000520) | 23.29 265 | | |
| (000520) | 23.29 | | (000520) 23.30 |
| (000522) | 24.01 270 | | (000517) 23.26 |
| (000524) | 24.03 275 | | |
| (000524) | 24.03 | | (000524) 24.04 |
| (000526) | 24.05 280 | | (000521) 23.30 |
| (000527) | 24.06 290 | | |
| (000527) | 24.06 | | (000527) 24.07 |
| (000530) | 24.09 5003 | | |
| (000534) | 24.11 291 | | (000532) 24.10 |
| (000539) | 24.12 292 | | (000532) 24.10 |
| (000544) | 24.13 293 | | (000532) 24.10 |
| (000549) | 24.14 294 | | (000532) 24.10 |
| (000553) | 24.15 298 | | (000537) 24.11 (000542) 24.12 (000547) 24.13 |
| (000559) | 24.18 60 | | |
| (000559) | 24.18 | | (000559) 24.20 |
| (000562) | 24.21 5004 | | (000529) 24.08 |
| (000566) | 24.23 321 | | (000564) 24.22 |
| (000567) | 24.24 325 | | |
| (000567) | 24.24 | | (000567) 24.25 |
| (000570) | 24.26 322 | | (000564) 24.22 |
| (000571) | 24.27 336 | | |
| (000571) | 24.27 | | (000571) 24.28 |
| (000574) | 24.29 323 | | (000584) 24.22 |
| (000575) | 24.30 326 | | |
| (000575) | 24.30 | | (000575) 24.31 |
| (000578) | 25.01 324 | | (000564) 24.22 |
| (000579) | 25.02 327 | | |
| (000579) | 25.02 | | (000579) 25.03 |
| (000581) | 25.04 300 | | (000487) 22.11 (000475) 22.13 (000481) 22.15 (000488) 23.05 (000568) 24 25 (000572) 24.28 (000576) 24.31 |

CHART TITLE - NON-PROCEDURAL STATEMENTS

CHART TITLE - INTRODUCTORY COMMENTS

CHART TITLE - FUNCTION TBL(XMACH,P0,T0,TSK,DIST,XLAMDA,REY,METHOD,IER,TOT)

| | | | |
|---|---|---|---|
| (000637) | 28.02 10 | | |
| (000639) | 29.01 20 | | (000636) 28.01 |
| (000640) | 29.02 30 | | |
| (000641) | 29.03 40 | | (000639) 29.01 |
| (000653) | 29.07 50 | | (000673) 29.19 |
| (000660) | 29.11 52 | | |
| (000661) | 29.12 54 | | |
| (000665) | 29.13 58 | | (000659) 29.10 |
| (000667) | 29.15 60 | | |
| (000668) | 29.16 70 | | (000666) 29.14 (000684) 29.20 |
| (000671) | 29.18 80 | | |
| (000672) | 29.19 85 | | |
| (000683) | 29.20 56 | | (000660) 29.11 |
| (000674) | 29.21 90 | | (000671) 29.18 |
| (000678) | 29.22 100 | | (000638) 28.02 (000662) 29.12 (000670) 29.17 |

CHART TITLE - INTRODUCTORY COMMENTS

CHART TITLE - FUNCTION TEHALT(ALT,TI)

| CARD ID | PAGE/BOX | NAME | REFERENCES (SOURCE SEQUENCE NO. AND PAGE/BOX) |
|---|---|---|---|
| (000718) | 31.01 | 200 | (000700) 31.07 |
| (000719) | 31.02 | 210 | |
| (000782) | 31.03 | 400 | (000701) 31.08 |
| (000783) | 31.04 | 410 | |
| (000698) | 31.06 | 10 | |
| (000700) | 31.07 | 20 | |
| (000701) | 31.08 | 30 | |
| (000702) | 31.09 | 40 | |
| (000705) | 31.10 | 100 | (000700) 31.07 |
| (000706) | 31.11 | 110 | |
| (000759) | 31.12 | 300 | (000701) 31.08 |
| (000760) | 31.13 | 310 | |
| (000708) | 31.14 | 120 | (000705) 31.10 |
| (000709) | 31.15 | 125 | (000705) 31.10 |
| (000763) | 31.16 | 320 | (000759) 31.12 |
| (000764) | 31.17 | 325 | (000759) 31.12 |
| (000711) | 32.01 | 140 | (000708) 31.14 |
| (000712) | 32.02 | 145 | |
| (000714) | 32.03 | 160 | (000711) 32.01 |
| (000721) | 33.01 | 220 | (000718) 31.01 |
| (000722) | 33.02 | 225 | (000718) 31.01 |
| (000724) | 33.03 | 230 | (000721) 33.01 |
| (000725) | 33.04 | 235 | |
| (000726) | 33.05 | 236 | |
| (000732) | 33.06 | 240 | (000724) 33.03 |
| (000733) | 33.07 | 245 | (000724) 33.03 |
| (000729) | 33.08 | 237 | (000725) 33.04 |
| (000735) | 33.09 | 250 | (000732) 33.06 |
| (000738) | 33.10 | 251 | |
| (000739) | 34.01 | 252 | (000735) 33.09 |
| (000740) | 34.02 | 253 | |
| (000742) | 34.03 | 260 | (000739) 34.01 |
| (000743) | 34.04 | 265 | (000739) 34.01 |
| (000745) | 34.05 | 270 | (000742) 34.03 |
| (000746) | 34.06 | 275 | |
| (000750) | 34.08 | 280 | (000745) 34.05 |
| (000751) | 34.09 | 285 | |
| (000754) | 35.01 | 290 | (000750) 34.08 |
| (000767) | 35.02 | 330 | (000763) 31.16 |
| (000768) | 35.03 | 335 | (000763) 31.16 |
| (000770) | 35.04 | 340 | (000767) 35.02 |
| (000771) | 35.05 | 345 | (000767) 35.02 |
| (000773) | 35.06 | 350 | (000770) 35.04 |
| (000774) | 35.07 | 360 | |
| (000777) | 35.08 | 370 | (000773) 35.06 |
| (000785) | 36.01 | 420 | (000782) 31.03 |
| (000786) | 36.02 | 425 | (000782) 31.03 |
| (000788) | 36.03 | 430 | (000785) 36.01 |
| (000789) | 36.04 | 435 | (000785) 36.01 |
| (000792) | 36.05 | 440 | (000788) 36.03 |
| (000794) | 36.06 | 445 | (000788) 36.03 |
| (000795) | 36.07 | 450 | (000792) 36.05 |
| (000799) | 37.01 | 500 | (000702) 31.09 |
| (000800) | 37.02 | 501 | |
| (000801) | 37.03 | 502 | |
| (000806) | 37.04 | 520 | (000799) 37.01 |
| (000807) | 37.05 | 525 | (000799) 37.01 |
| (000808) | 37.06 | 526 | |
| (000803) | 37.07 | 503 | (000800) 37.02 |
| (000814) | 37.08 | 530 | (000806) 37.04 |
| (000811) | 37.09 | 527 | (000807) 37.05 |
| (000819) | 38.01 | 539 | (000702) 31.09 |
| (000821) | 38.03 | 540 | |
| (000822) | 38.04 | 542 | |
| (000826) | 38.05 | 550 | (000820) 38.02 |
| (000828) | 38.06 | 600 | (000720) 31.02 (000784) 31.04 (000707) 31.11 (000762) 31.13 (000766) 31.17<br>(000723) 33.02 (000728) 33.05 (000734) 33.07 (000731) 33.08 (000738) 33.10<br>(000741) 34.02 (000744) 34.04 (000749) 34.07 (000753) 34.09 (000756) 35.01 |

```
CARD ID   PAGE/BOX   NAME                      REFERENCES  (SOURCE SEQUENCE NO. AND PAGE/BOX)

                               (000769)  35.03    (000772)  35.05    (000776)  35.07    (000779)  35.08    (000787)  36.02
                               (000791)  36.04    (000794)  36.06    (000796)  36.07    (000802)  37.03    (000810)  37.06
                               (000805)  37.07    (000818)  37.08    (000813)  37.09    (000823)  38.04    (000825)  38.09

(000829)   38 07  810          (000715)  32.04

(000831)   38.08  830          (000710)  31.15    (000713)  32.02    (000715)  32.04    (000828)  38.06

(000824)   38.09  543          (000821)  38.03

CHART TITLE - NON-PROCEDURAL STATEMENTS


CHART TITLE - INTRODUCTORY COMMENTS


CHART TITLE - SUBROUTINE  TEMPER(XMACH,ALT,PRESSI,TLOC,TTOT,SUN,XSKIN,XSKINF,IER

(000840)   41.01  TEMPER       (000081)   3.01-X  (001547)  84.14-X  (001554)  84.18-X  (001563)  84.21-X

(000890)   42.02  59           (000952)  42.12

(000952)   42.12  103          (000941)  42  .    (000944)  43.01

(000956)   42.13  105          (000952)  42.12    (000947)  43.02

(000943)   43.01  101          (000941)  42.11

(000946)   43 02  102          (000941)  42.11

CHART TITLE - NON-PROCEDURAL STATEMENTS


CHART TITLE - INTRODUCTORY COMMENTS


CHART TITLE - FUNCTION  TSKIN(A,B,C,IER)

(000978)   46 02  2

(000979)   46.03  3

(000987)   46.05  7            (000977)  46.01

(000988)   46 06  8

(000983)   46.08  5            (000978)  46.02

(000994)   46 10  15           (000987)  46.05

(001001)   46.12  21           (001017)  46.18

(001004)   46 13  25

(001005)   46.14  30

(001007)   46.[illegible]  [illegible]     (001004)  46.13

(001012)   46.16  45

(001013)   46.17  50

(001018.   48.19  55

(0010.9)   46.2(  60           (001012)  46.16

(001020)   46.21  70           (000902)  46.04    (000991)  46.07    (000906)  46.09    (001006)  46.14    (000993)  47.01

(000952)   47.01  10           (000987)  46.05

CHART TITLE - NON-PROCEDURAL STATEMENTS


CHART TITLE - INTRODUCTORY COMMENTS


CHART TITLE - FUNCTION  TTO(XMACH,TO,IER)

(001055)   50.02  3

(001057)   50.03  5            (001054)  50.01

(001066)   50.08  10           (001075)  50.10

(001073)   50.09  20

(:01074)   50 10  30

(001076)   50.11  40           (001073)  50.09

(001078)   50 12  50           (001056)  5C.02    (001072)  50.08

CHART TITLE - INTRODUCTORY COMMENTS


CHART TITLE - SUBROUTINE  WHVMAT

(00108C.   52.01  WHVMAT       (001297)  59 01-X

(00111.)   52.04               (001247)  56.27

(001113)   52.05  601          (001111)  52.04

(001115)   52.06  605          (001113)  52.05

(001118)   52.07  602          (001111)  52.04

(001120)   52.08  615          (001118)  52.07

(001121)   52.09  616

(001124)   52.10  603          (001111)  52.04

(001125)   53.01  625          (001124)  52.10

(001127)   53.02  626
```

| CARD ID | PAGE/BOX | NAME | REFERENCES (SOURCE SEQUENCE NO. AND PAGE/BOX) | | | | |
|---|---|---|---|---|---|---|---|
| (0011291) | 53 03 | 610 | (0011181) 52 06 | (0011221) 52 09 | | | |
| (0011301) | 53 04 | 611 | | | | | |
| (0011301) | 53 04 | | (0011301) 53 05 | | | | |
| (0011351) | 53 08 | | (0011431) 53 12 | | | | |
| (0011401) | 53 10 | 638 | | | | | |
| (0011421) | 53 11 | 639 | (0011391) 53 09 | | | | |
| (0011431) | 53 12 | 640 | (0011411) 53 10 | | | | |
| (0011511) | 53 14 | 801 | (0011491) 53 13 | | | | |
| (0011551) | 54 01 | 805 | (0011491) 53 13 | | | | |
| (0011581) | 54 03 | | (0011611) 54.05 | | | | |
| (0011591) | 54 04 | 807 | | | | | |
| (0011611) | 54 05 | 810 | (0011581) 54 03 | | | | |
| (0011631) | 54 06 | 815 | (0011601) 54 04 | | | | |
| (0011651) | 54 07 | 820 | (0011631) 54 06 | | | | |
| (0011691) | 55 01 | 825 | (0011631) 54 06 | | | | |
| (0011751) | 55 02 | 835 | (0011691) 55 01 | | | | |
| (0011761) | 55 03 | | (0011801) 55 05 | | | | |
| (0011771) | 55 04 | 836 | | | | | |
| (0011801) | 55 05 | 840 | (0011761) 55 03 | | | | |
| (0011831) | 55 06 | 845 | (0011671) 54 07 | (0011791) 55 04 | (0011731) 55 16 | | |
| (0011851) | 55 07 | 850 | (0011531) 53 14 | | | | |
| (0011871) | 55 08 | 852 | (0011201) 52 08 | (0011261) 53 01 | | | |
| (0011891) | 55 09 | 5001 | | | | | |
| (0011901) | 55 11 | 669 | | | | | |
| (0011931) | 55 13 | 668 | (0011851) 55 10 | | | | |
| (0011971) | 55 15 | 671 | (0011951) 55 14 | | | | |
| (0011711) | 55 16 | 830 | (0011691) 55 01 | | | | |
| (0012011) | 55 17 | 672 | (0011951) 55 14 | | | | |
| (0012051) | 56 01 | 671 | (0011951) 55 14 | | | | |
| (0012091) | 56 02 | 650 | (0011991) 55 15 | (0012031) 55 17 | | | |
| (0012141) | 56 05 | 50 | | | | | |
| (0012141) | 56 05 | | (0012141) 56 07 | | | | |
| (0012201) | 56 10 | 5002 | (0011871) 55 08 | | | | |
| (0012241) | 56 12 | 681 | (0012221) 56 11 | | | | |
| (0012251) | 56 13 | | (0012271) 56 15 | | | | |
| (0012271) | 56 14 | 691 | | | | | |
| (0012321) | 56 17 | 682 | (0012221) 56 11 | | | | |
| (0012331) | 56 18 | | (0012351) 56 20 | | | | |
| (0012351) | 56 19 | 692 | | | | | |
| (0012401) | 56 22 | 683 | (0012221) 56 11 | | | | |
| (0012411) | 56 23 | | (0012431) 56 25 | | | | |
| (0012431) | 56 24 | 693 | | | | | |
| (0012471) | 56 27 | 600 | (0011131) 52 05 | (0011181) 52 07 | (0011241) 52.10 | (0012301) 56.16 | (0012381) 56.21 |

CHART TITLE - NON PROCEDURAL STATEMENTS

CHART TITLE - INTRODUCTORY COMMENTS

CHART TITLE - SUBROUTINE HWVOO

| CARD ID | PAGE/BOX | NAME | REFERENCES | |
|---|---|---|---|---|
| (0012571) | 59 01 | HWVOO | (0000951) 3 06-X | |
| (0013071) | 59 05 | | (0013081) 59 06 | |
| (0013081) | 59 06 | 10 | | |
| (0013131) | 59 08 | 12 | | |
| (0013141) | 59 09 | 14 | (0013121) 59 07 | |
| (0013171) | 59 11 | 20 | | |
| (0013221) | 59 13 | 33 | (0013161) 59 10 | |
| (0013291) | 59 18 | | (0013341) 59 19 | |
| (0013341) | 59 19 | 35 | | |
| (0013411) | 59 20 | 99 | (0013981) 60 25 | |
| (0013431) | 59 21 | | (0013841) 60.17 | |
| (0013451) | 59 22 | 100 | (0013441) 59 21 | |
| (0013461) | 59 23 | 102 | | |
| (0013471) | 59 24 | | (0013481) 59 25 | |
| (0013481) | 59 25 | 110 | | |
| (0013491) | 59 26 | 115 | (0013811) 60 04 | (0013681) 60 08 |
| (0013511) | 59 28 | | (0013551) 59 30 | |
| (0013521) | 59 29 | 122 | | |
| (0013551) | 59 30 | 120 | (0013511) 59.28 | |
| (0013571) | 60 01 | 200 | (0013441) 59 21 | |
| (0013581) | 60 02 | 202 | | |

| CARD ID | PAGE/BOX | | NAME | REFERENCES (SOURCE SEQUENCE NO. AND PAGE/BOX) | | | |
|---|---|---|---|---|---|---|---|
| (001359) | 60.03 | | | (001360) 60.04 | | | |
| (001360) | 60 04 | 210 | | | | | |
| (001362) | 60 05 | 300 | | (001344) 59 21 | | | |
| (001363) | 60.06 | 302 | | | | | |
| (001364) | 60 07 | | | (001365) 60.08 | | | |
| (001365) | 60.08 | 310 | | | | | |
| (001367) | 60.09 | 400 | | (001356) 59.30 | | | |
| (001368) | 60.10 | 402 | | | | | |
| (001369) | 60 11 | 410 | | (001368) 60.10 | | | |
| (001372) | 60 12 | 420 | | (001368) 60 10 | | | |
| (001375) | 60 13 | 430 | | (001368) 60.10 | | | |
| (001377) | 60 14 | 440 | | (001371) 60 11 | (001374) 60.12 | | |
| (001384) | 60.17 | 500 | | (001345) 59.22 | (001357) 60.01 | (001362) 60 05 | (001367) 60.09 |
| (001388) | 60.19 | 510 | | | | | |
| (001390) | 60 21 | 512 | | | | | |
| (001393) | 60.23 | 520 | | (001389) 60 20 | | | |
| (001394) | 60 24 | | | (001395) 60.25 | | | |
| (001395) | 60 25 | 530 | | | | | |
| (001400) | 61.01 | 600 | | (001387) 60 18 | | | |
| (001401) | 61 02 | 502 | | | | | |
| (001402) | 61 03 | 604 | | | | | |
| (001410) | 61 07 | 610 | | (001401) 61.02 | | | |
| (001413) | 61.09 | | | (001418) 61.13 | | | |
| (001418) | 61 13 | 620 | | | | | |
| (001420) | 61.15 | | | (001421) 61.16 | | | |
| (001421) | 61.16 | 630 | | | | | |
| (001424) | 61.18 | 640 | | (001439) 61.31 | | | |
| (001426) | 61 20 | | | (001430) 61.22 | | | |
| (001427) | 61.21 | 644 | | | | | |
| (001430) | 61.22 | 642 | | (001426) 61.20 | | | |
| (001432) | 61 24 | 650 | | | | | |
| (001433) | 61.25 | 652 | | | | | |
| (001435) | 61.28 | 660 | | (001432) 61.24 | | | |
| (001437) | 61.30 | | | (001438) 61.31 | | | |
| (001438) | 61.31 | 662 | | | | | |
| (001440) | 62.01 | 670 | | (001431) 61.23 | | | |
| (001441) | 62 02 | 672 | | | | | |
| (001448) | 62 07 | 700 | | (001400) 61.01 | (001440) 62.01 | | |
| (001449) | 62 08 | 702 | | | | | |
| (001459) | 62.12 | | | (001461) 62.13 | | | |
| (001461) | 62 13 | 710 | | | | | |
| (001465) | 62.14 | 712 | | (001530) 63.29 | | | |
| (001466) | 62.15 | | | (001508) 63.15 | | | |
| (001471) | 62 17 | 714 | | | | | |
| (001473) | 62.18 | 716 | | (001470) 62.16 | | | |
| (001474) | 62 19 | | | (001478) 62.21 | | | |
| (001476) | 62 21 | 718 | | | | | |
| (001477) | 62 22 | 72. | | (001472) 62.17 | | | |
| (001479) | 62 23 | 722 | | (001475) 62.20 | | | |
| (001484) | 63.01 | 724 | | (001470) 62.18 | | | |
| (001485) | 63 02 | 726 | | (001475) 62.20 | | | |
| (001488) | 63.03 | 728 | | (001482) 62 24 | | | |
| (001489) | 63.04 | 730 | | | | | |
| (001491) | 63 05 | [illegible] | | (001488) 63.03 | | | |
| (001492) | 63.06 | 734 | | | | | |
| (001494) | 63.07 | 736 | | (001488) 63.03 | (001491) 63.05 | | |
| (001495) | 63 08 | 738 | | (001491) 63.05 | | | |
| (001497) | 63 09 | 740 | | (001490) 63.04 | (001493) 63.06 | | |
| (001498) | 63.10 | 742 | | | | | |
| (001501) | 63.12 | 746 | | (001497) 63 09 | | | |
| (001503) | 63.14 | 748 | | | | | |
| (001508) | 63.15 | 760 | | (001502) 63.13 | | | |
| (001513) | 63.17 | 762 | | | | | |
| (0015.4) | 63.18 | 764 | | | | | |
| (001526) | 63.26 | 772 | | (001513) 63 17 | | | |
| (001528) | 63.28 | | | (001529) 63.29 | | | |
| (001529) | 63.29 | 774 | | | | | |
| (001531) | 64.01 | 780 | | (001512) 63.16 | | | |
| (001532) | 64.02 | 782 | | | | | |
| (001543) | 64 11 | 800 | | (001448) 62.07 | (001531) 64.01 | | |
| (001544) | 64.12 | | | (001548) 64.15 | | | |
| (001545) | 64.13 | 812 | | | | | |

| CARD ID | PAGE/BOX | NAME | REFERENCES (SOURCE SEQUENCE NO. AND PAGE/BOX) | | | | |
|---|---|---|---|---|---|---|---|
| (001548) | 64 15 | 810 | (001544) 64 12 | | | | |
| (001553) | 64 17 | 820 | | | | | |
| (001559) | 64 19 | 850 | (001552) 64 18 | | | | |
| (001560) | 64 20 | 852 | | | | | |
| (001570) | 64 22 | 899 | (001559) 64 19 | | | | |
| (001577) | 64 24 | | (001661) 68 02 | | | | |
| (001583) | 64 25 | 901 | (001581) 64 24 | | | | |
| (001585) | 64 26 | 921 | (001583) 64 25 | | | | |
| (001586) | 64 27 | | (001587) 64 29 | | | | |
| (001587) | 64 28 | 905 | | | | | |
| (001592) | 64 31 | 902 | (001581) 64 24 | | | | |
| (001594) | 65 01 | 922 | (001592) 64 31 | | | | |
| (001595) | 65 02 | | (001596) 65 04 | | | | |
| (001596) | 65 03 | 906 | | | | | |
| (001601) | 65 06 | 903 | (001581) 64 24 | (001581) 64 24 | | | |
| (001603) | 65 07 | 923 | (001601) 65 06 | | | | |
| (001604) | 65 08 | | (001605) 65 10 | | | | |
| (001605) | 65 09 | 907 | | | | | |
| (001608) | 65 13 | 908 | | | | | |
| (001610) | 65 14 | 909 | (001607) 65 12 | | | | |
| (001611) | 65 15 | 916 | | | | | |
| (001613) | 66 01 | 904 | (001581) 64 24 | | | | |
| (001614) | 66 02 | 931 | | | | | |
| (001615) | 66 03 | 932 | | | | | |
| (001618) | 66 04 | | (001617) 66 06 | | | | |
| (001617) | 66 05 | 935 | | | | | |
| (001621) | 66 08 | 910 | (001590) 64 30 | (001599) 65 05 | (001609) 65.13 | (001612) 65.15 | |
| (001623) | 66 09 | 952 | (001621) 66 08 | | | | |
| (001626) | 66 10 | 955 | (001621) 66 08 | | | | |
| (001628) | 66 11 | 960 | (001626) 66 10 | | | | |
| (001632) | 67 01 | 965 | (001626) 66 10 | | | | |
| (001638) | 67 02 | 975 | (001632) 67 01 | | | | |
| (001639) | 67 03 | | (001643) 67 05 | | | | |
| (001640) | 67 04 | 976 | | | | | |
| (001643) | 67 05 | 982 | (001639) 67 03 | | | | |
| (00164[illegible]) | 67 06 | 985 | (001630) 66 11 | (001642) 67 04 | (001636) 67.09 | | |
| (00164[illegible]) | 67 07 | 990 | (001624) 66 09 | | | | |
| (00164[illegible]) | 67 08 | 911 | (001647) 67 07 | | | | |
| (00163[illegible]) | 67 09 | 980 | (001632) 67 01 | | | | |
| (00165[illegible]) | 67 10 | 912 | (001647) 67 07 | | | | |
| (001655) | 67 11 | 913 | (001647) 67 07 | | | | |
| (001658) | 67 12 | 914 | (001647) 67 07 | | | | |
| (001660) | 68 01 | 915 | (001647) 67 07 | | | | |
| (001661) | 68 02 | 900 | (001583) 64 25<br>(001614) 66 02 | (001592) 64 31<br>(001650) 67 08 | (001601) 65 06<br>(001653) 67.10 | (001610) 65.14<br>(001656) 67.11 | (001613) 66 01<br>(001659) 67.12 |
| (001672) | 68 05 | 96[illegible] | (001670) 68 04 | | | | |
| (001673) | 68 06 | 9[illegible]1 | | | | | |
| (001676) | 68 07 | 997 | (001672) 68 05 | | | | |
| (001678) | 68 08 | 1005 | (001675) 68 06 | | | | |
| (001680) | 68 11 | 1000 | (001670) 68 04 | | | | |

CHART TITLE - NON-PROCEDURAL STATEMENTS

| LOCATION | | DIAGNOSTIC |
|---|---|---|
| CARD ID | PAGE/BOX | |
| (000024) | 2.08 | UNDEFINED - 'READMS' EXTERNAL REFERENCE |
| (000025) | 2.09 | UNDEFINED - 'READMS' EXTERNAL REFERENCE |
| (000098) | 3.07 | UNDEFINED - 'WRITMS' EXTERNAL REFERENCE |
| (001132) | 53.06 | UNDEFINED - 'READMS' EXTERNAL REFERENCE |
| (001302) | 59.02 | UNDEFINED - 'READMS' EXTERNAL REFERENCE |
| (001449) | 62.08 | UNDEFINED - 'READMS' EXTERNAL REFERENCE |
| (001689) | 68.11 | UNDEFINED - 'WRITMS' EXTERNAL REFERENCE |

# PROGRAM FLOW CHARTS

# OF

# FLUTTER AND TEMPERATURE MODULE

CHART TITLE - PROCEDURES

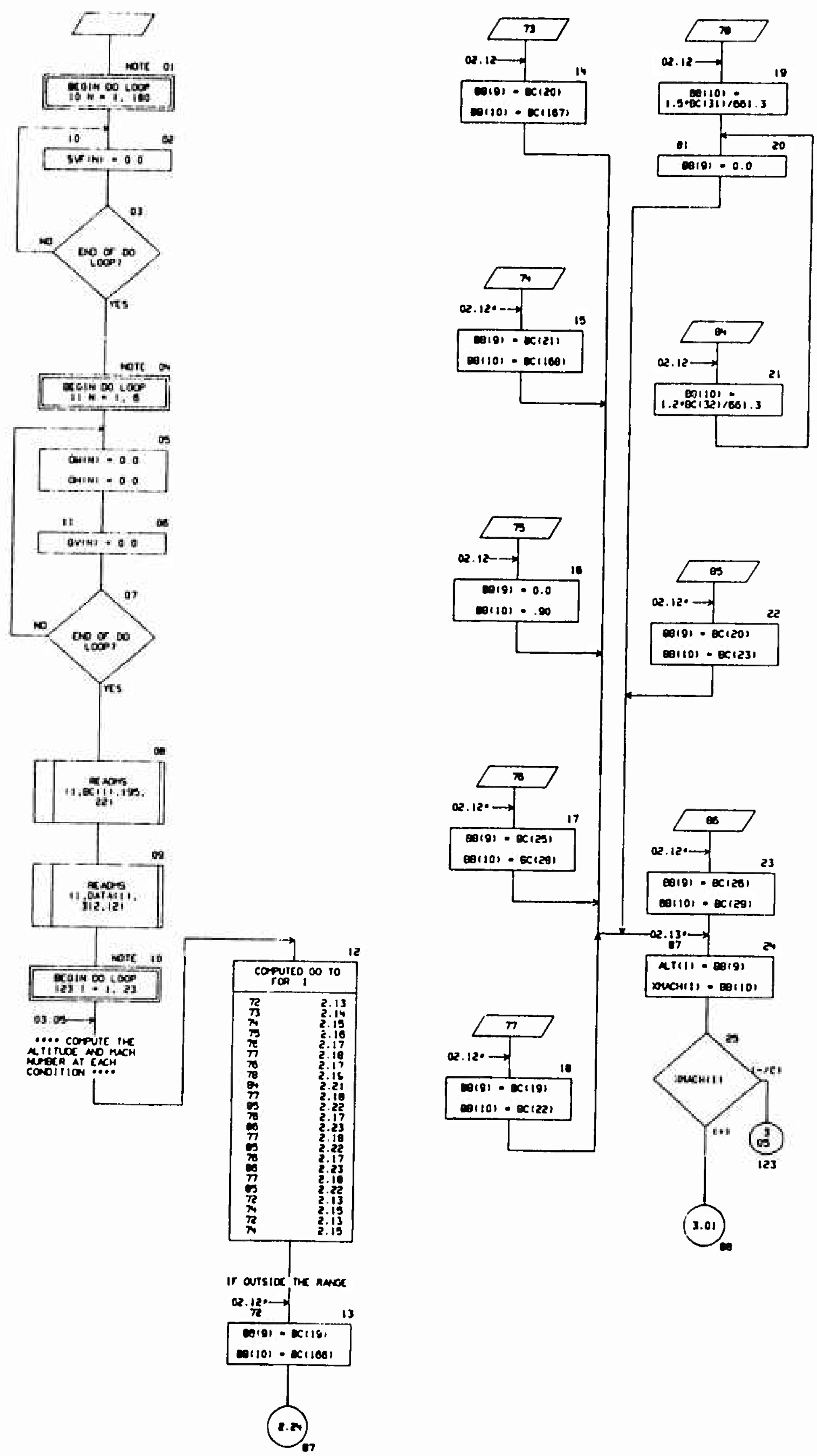
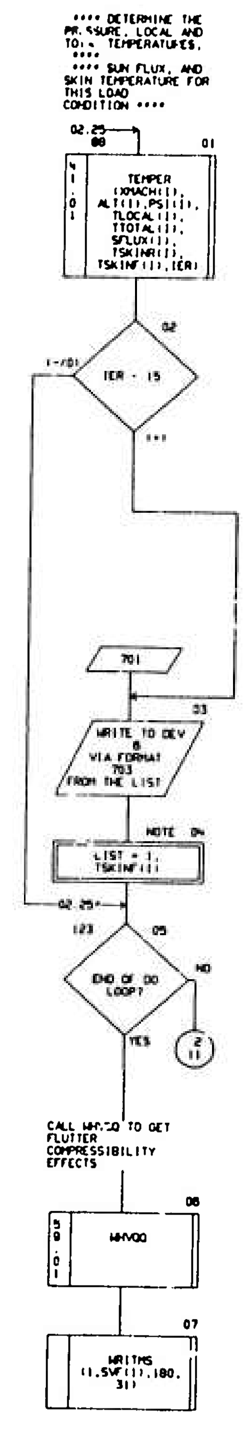


CHART TITLE - PROCEDURES



```
        PROGRAM OLAY3
        COMMON    SVF(180),DATA(312),TFM(38),TFH(38),TFV(38),TFWF(38),
                  TF(38),XMN1(20),Q(20),QQ(20),XMATOP(20),
                  OH(8),OH(8),OV(8)
        DIMENSION PS(23),TLOCAL(23),TTOTAL(23),SFLUX(23),TSKINR(23),
                  TSKINF(23)
        DIMENSION BC(195),ALT(23),XMACH(23),BB(10)
        EQUIVALENCE         (SVF(  1),PS    (1)),(SVF( 24),TLOCAL(1)),
         (SVF( 47),TTOTAL(1)),(SVF( 70),SFLUX (1)),(SVF( 93),TSKINR(1)),
         (SVF(116),TSKINF(1))
703     FORMAT(15HOLOAD CONDITION,I3,77H - THE TEMPERATURE LOOP DID NOT CL
        OSE IN 100 ITERATIONS. THE SKIN TEMPERATURE,F8.3,27H IS FROM THE L
        AST ITERATION/)
```

CHART TITLE - INTRODUCTORY COMMENTS

```
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
                        FUNCTION HBL
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''

   NOTE-  THIS FUNCTION SUBPROGRAM COMPUTES AERODYNAMIC HEATING
   COEFFICIENTS BY THE METHOD OF E R. VAN DRIEST.  IT IS DESCRIBED IN
   REPORT NA-63-1473.  THIS VERSION IS IN FORTRAN IV.


   PROGRAM TO COMPUTE BOUNDARY LAYER HEAT TRANSFER COEFFICIENTS

   C.J. MAC MILLER                          26 MARCH,1964

   THE EQUATION IS PROGRAMMED AS FOLLOWS....

     BOUNDARY LAYER HEAT TRANSFER COEFFICIENT  =

          HBL(XMACH,PO,TO,TSK,DIST,XLAMDA,REY,METHOD,IER)

      WHERE - XMACH = LOCAL MACH NUMBER, DIMENSIONLESS
                 PO = LOCAL PRESSURE, PSF
                 TO = LOCAL TEMPERATURE, DEG RANKINE
                TSK = SKIN TEMPERATURE, DEG RANKINE
               DIST = CHARACTERISTIC LENGTH, FEET
                      WITH METHOD 1-2, = LENGTH AFT OF LEADING EDGE
                      WITH METHOD 3-4, = DIAMETER OF LEADING EDGE
             XLAMDA = ANGLE OF SWEEP, DEGREES
                REY = TRANSITION REYNOLDS NUMBER, DIMENSIONLESS. IF
                      NO VALUE OF REY IS SUBMITTED, THE FUNCTION
                      WILL SUPPLY A VALUE OF ONE MILLION
             METHOD = SURFACE ORIENTATION
                      METHOD = 1 FOR FLAT PLATE, WEDGES, CYLINDERS
                                 ALIGNED WITH FLOW
                             = 2 FOR CONES OR OTHER SURFACES OF REV-
                                 OLUTION
                             = 3 FOR STAGNATION HEAT TRANSFER OF TWO
                                 DIMENSIONAL SHAPES AS CYLINDERS
                             = 4 FOR STAGNATION HEAT TRANSFER OF
                                 THREE DIMENSIONAL SHAPES AS SPHERES
                IER = ERROR INDICATOR
                 IF IER = 0, NO ERROR, SOLUTION CRITERIA WERE MET.
                    IER = 1, SOLUTION FOR TOTAL TEMPERATURE DID NOT
                             CONVERGE IN 100 PASSES, VALUE RETURNED
                             IS BASED ON CONSTANT GAMMA.
                    IER - 2, SOLUTION FOR BOUNDARY LAYER TEMPERATURE
                             DID NOT CONVERGE IN 100 PASSES, VALUE
                             RETURNED IS BASED ON RECOVERY OF .875
                 IF IER = 3, SOLUTION FOR SKIN FRICTION COEFFICIENT
                             DID NOT CONVERGE IN 100 PASSES, SOLU-
                             TION FOR BOUNDARY LAYER COEF. WAS CON-
                             TINUED WITH CURRENT VALUE OF C(F).
                    IER = 4, ARGUMENT XMACH IS ZERO, DIST IS ZERO,
                             HBL = 6.00
                    IER = 5, ARGUMENT XMACH IS ZERO, DIST IS
                             POSITIVE, HBL = 6.00
                    IER = 6, ARGUMENT XMACH IS POSITIVE, DIST IS
                             MINUS OR ZERO, HBL = 6.00
                   IER = 7, ARGUMENT TO IS BELOW 160 DEGREES RANKINE
                   A VALUE OF HBL =6.0 IS RETURNED.
                   IER = 8, ARGUMENT TSK IS BELOW 160 RANKINE.
                   SOLUTION CONTINUES USING TO FOR TSK.
```

CHART TITLE - INTRODUCTORY COMMENTS

THE VALUE OF BOUNDARY LAYER HEAT TRANSFER COEFFICIENT IS RETURNED
IN BTU/HR-FT**2-DEG F TO THE CALLING PROGRAM

****  ***  ****

CHART TITLE - FUNCTION HBL(XMACH,PO,TO,TSK,DIST,XLAMDA,REY,METHOD,IER,TOTI,TAHI

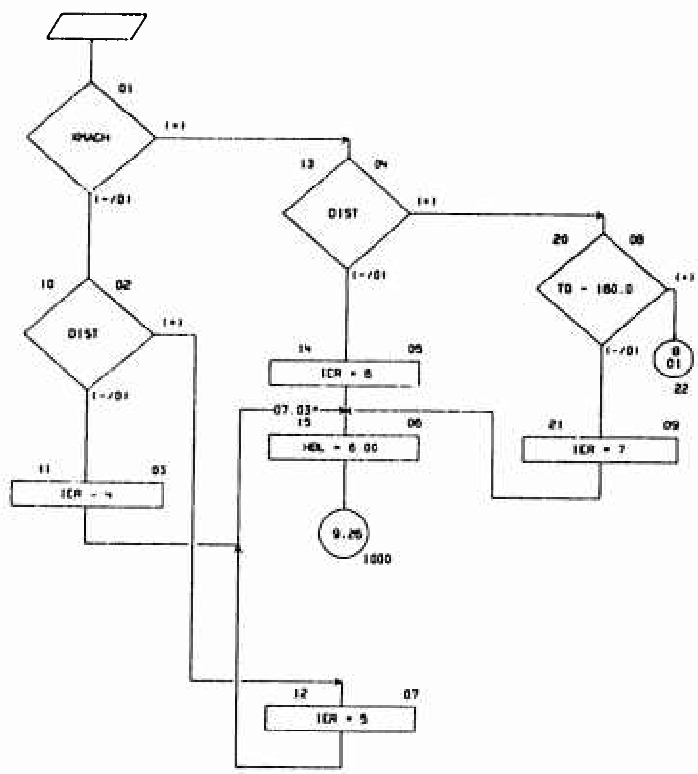


CHART TITLE - FUNCTION HBL(XMACH,PO,TO,TSK,DIST,XLAMDA,REY,METHOD,IER,TOTI,TAWI)

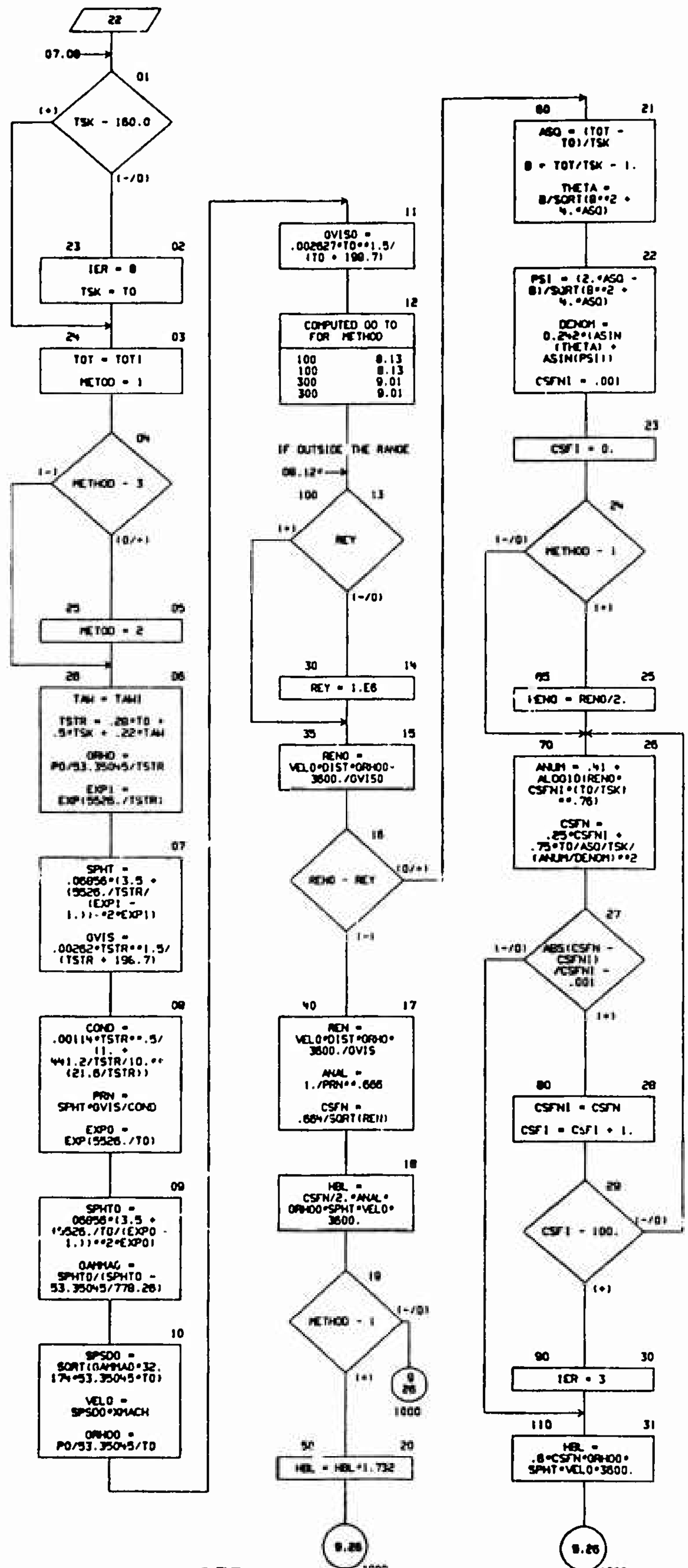


CHART TITLE - FUNCTION HBL(XMACH,PU TO,TSK,DIST,XLAMDA,REY,METHOD,IER,TOTI,TAHI

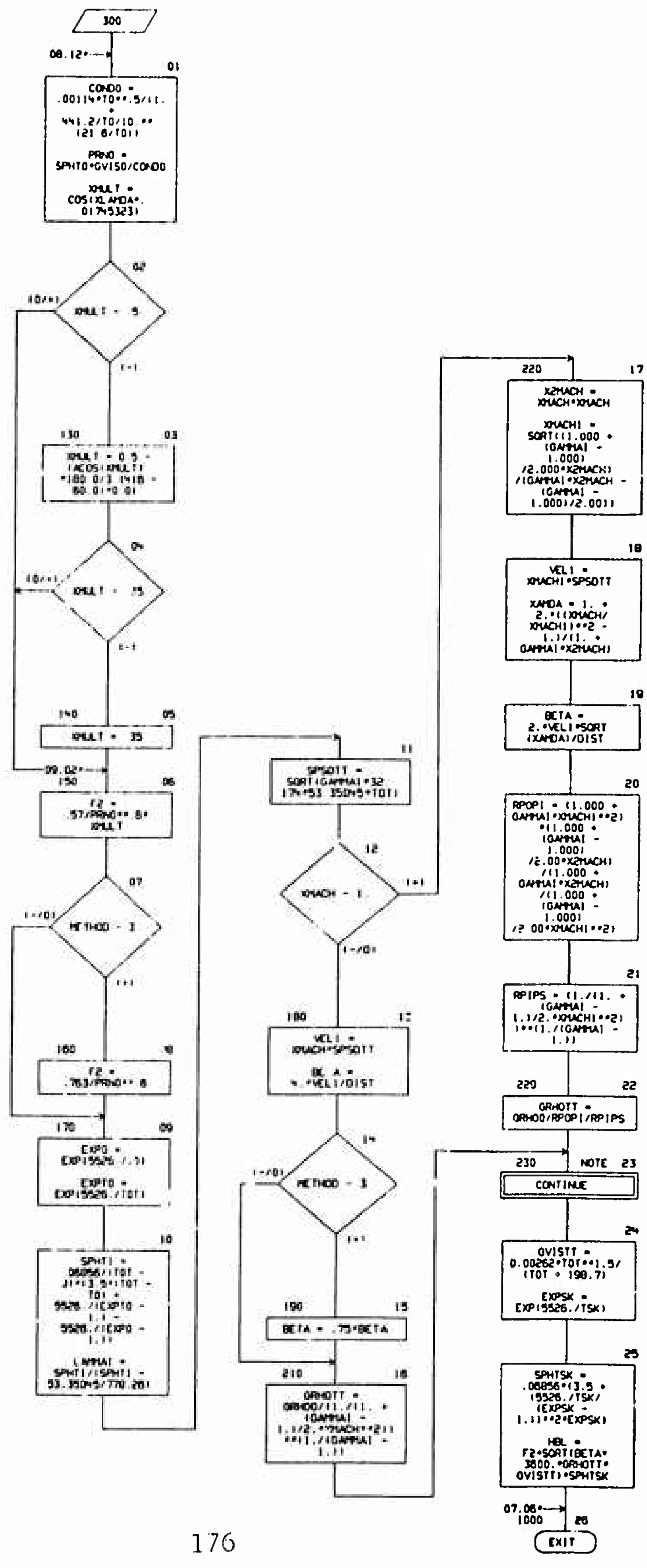


CHART TITLE - INTRODUCTORY COMMENTS

```
))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))
                            FUNCTION PRESH
((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((
```

CHART TITLE - FUNCTION PRESH(ALT)

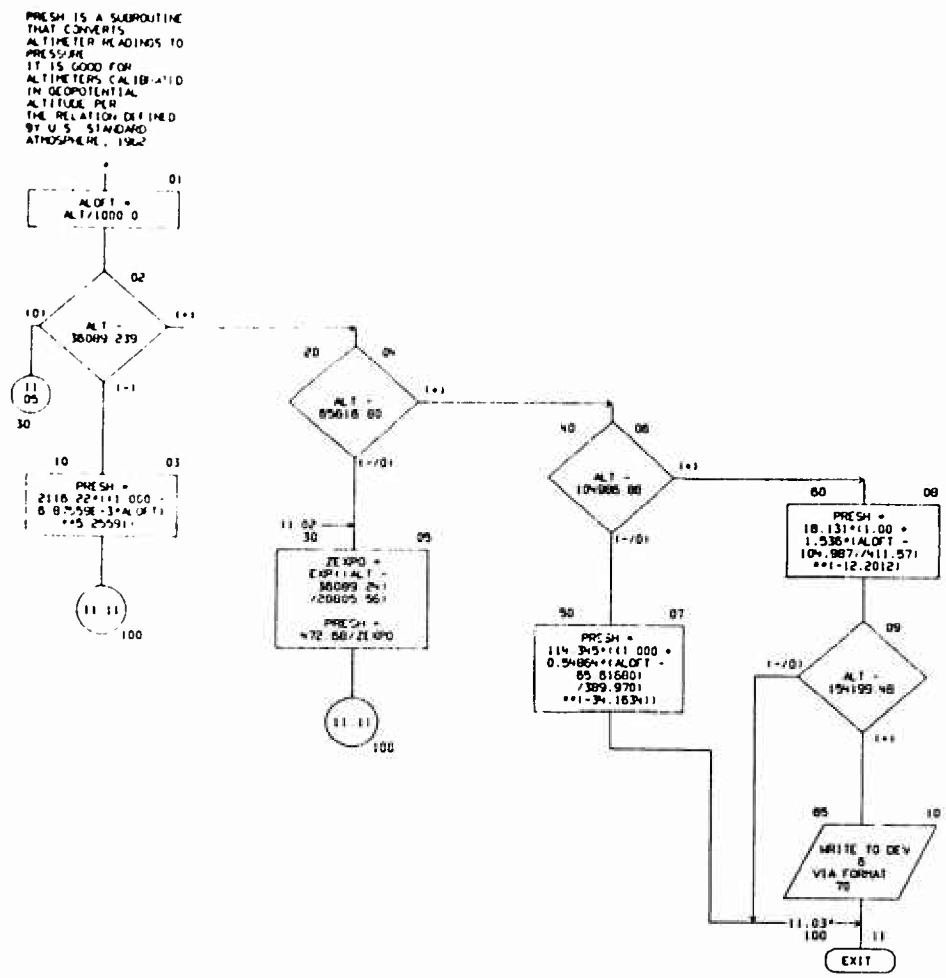


CHART TITLE - NON-PROCEDURAL STATEMENTS

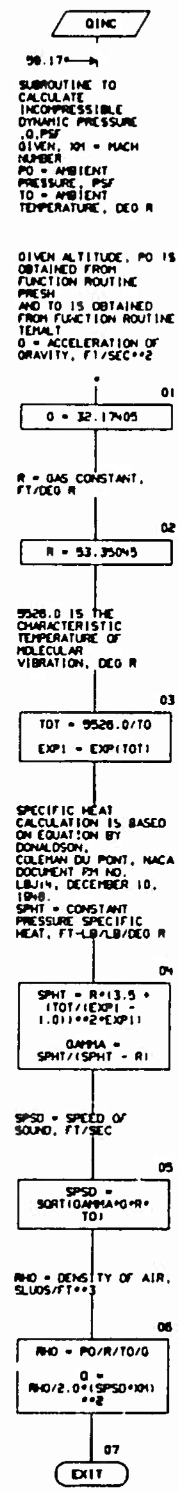
```
70      FORMAT(1H 8X49HWARNING- ALTITUDE IS BEYOND VALID RANGE OF PRESH.)
```

''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''

SUBROUTINE QINC

''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''

CHART TITLE - SUBROUTINE QINC(XM,PO,TO,Q)



SUBROUTINE QSUB

CHART TITLE - SUBROUTINE OSUB(LIM)

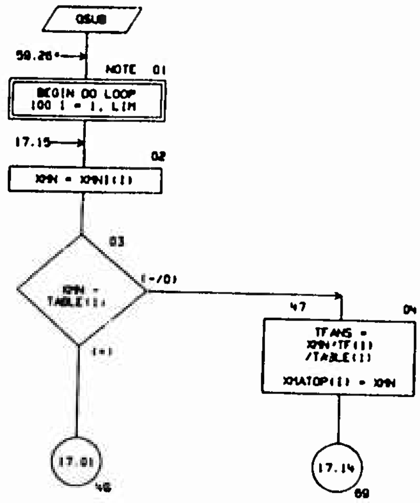


CHART TITLE - SUBROUTINE OSUBIL(N)

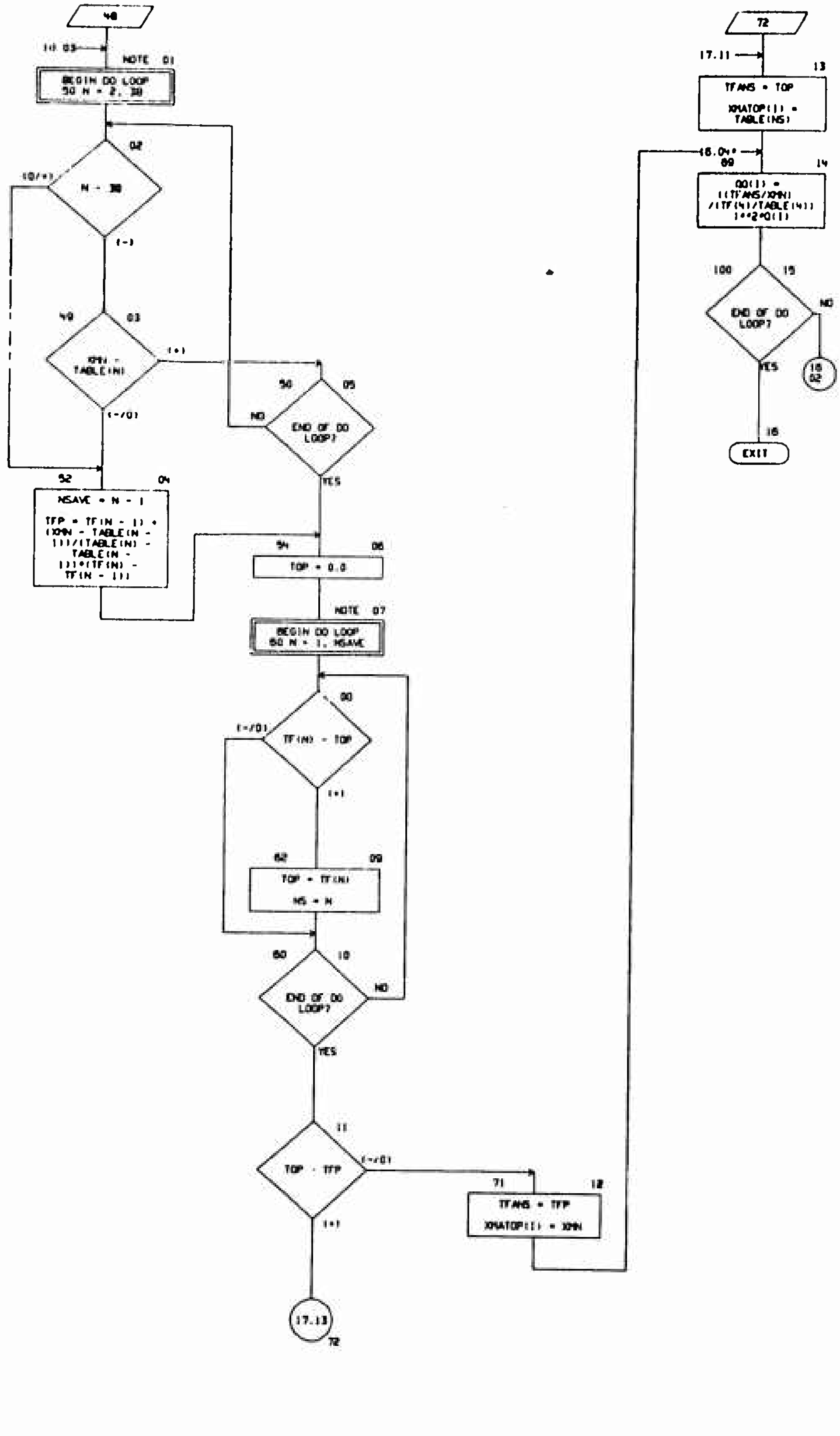


CHART TITLE - NON-PROCEDURAL STATEMENTS

```
COMMON    SVF(180),DATA(312),TFW(30),TFH(30),TFV(30),TFHF(30),
          TF(30),XMN1(20),Q(20),DQ(20),XMATOP(20)
DIMENSION TABLE(30)
EQUIVALENCE (TABLE(1),DATA(1))
```

```
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
FUNCTION SOLARO
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
```

CHART TITLE - FUNCTION SOLARO(ALT)

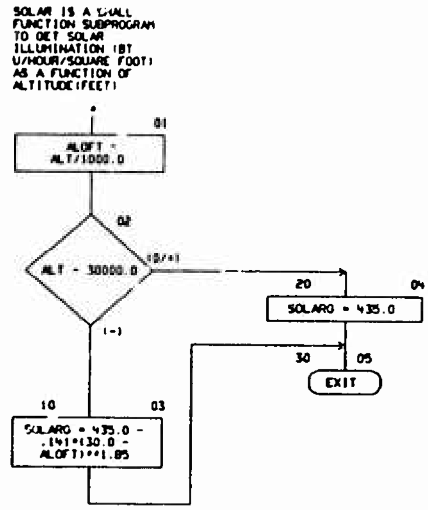


CHART TITLE - INTRODUCTORY COMMENTS

SUBROUTINE SVFTAD

CHART TITLE - SUBROUTINE SWTAB

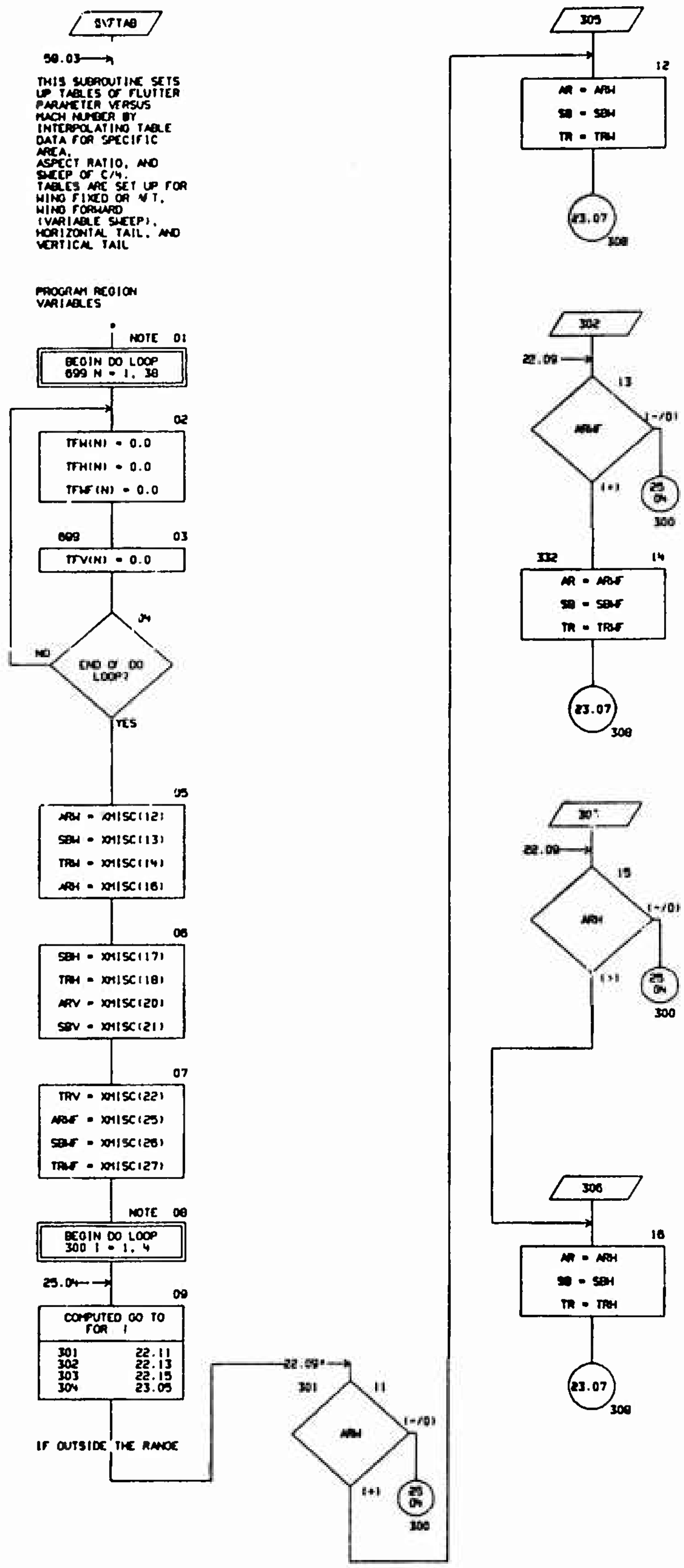


CHART TITLE - SUBROUTINE SWTAB

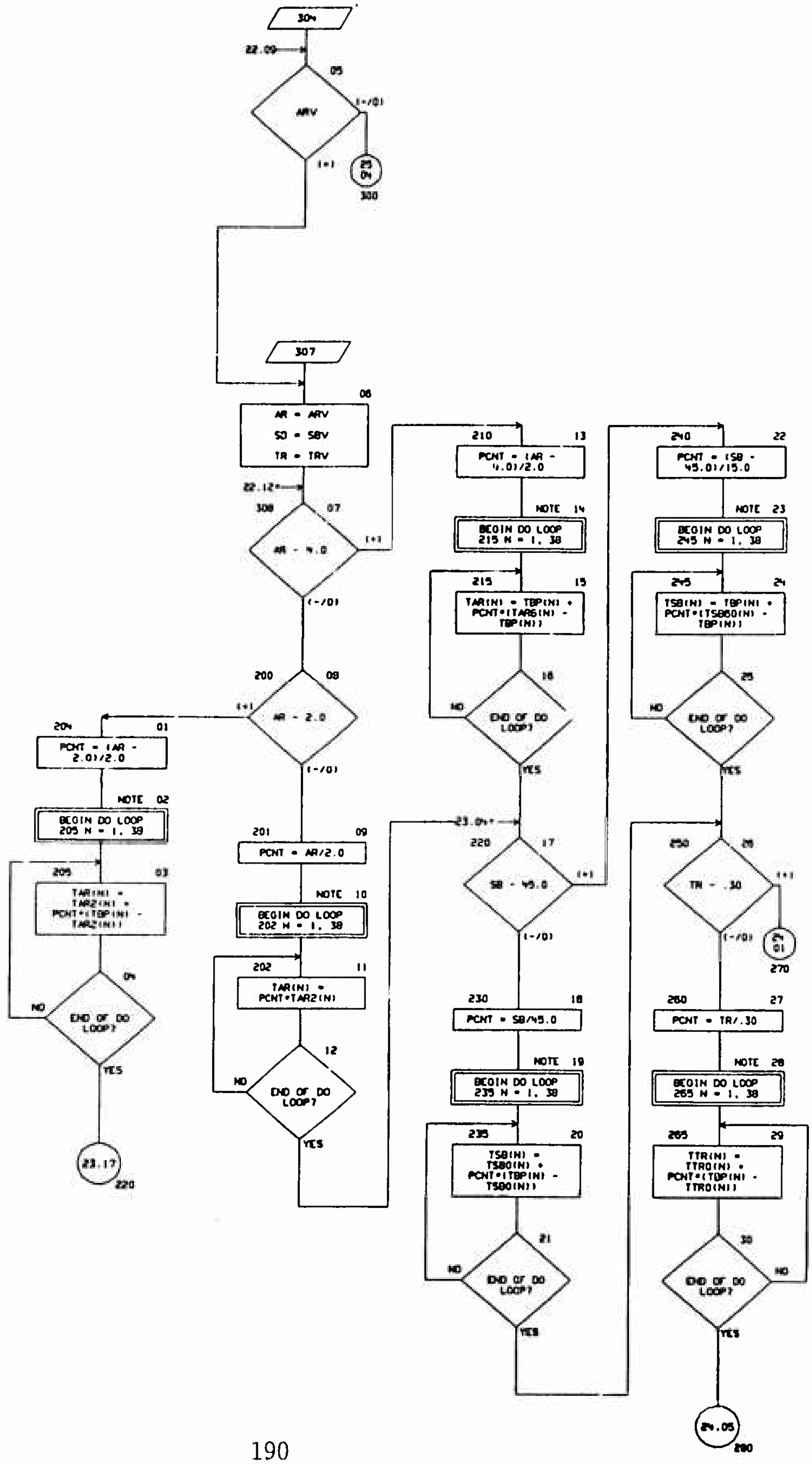


CHART TITLE - SUBROUTINE SVFTAB

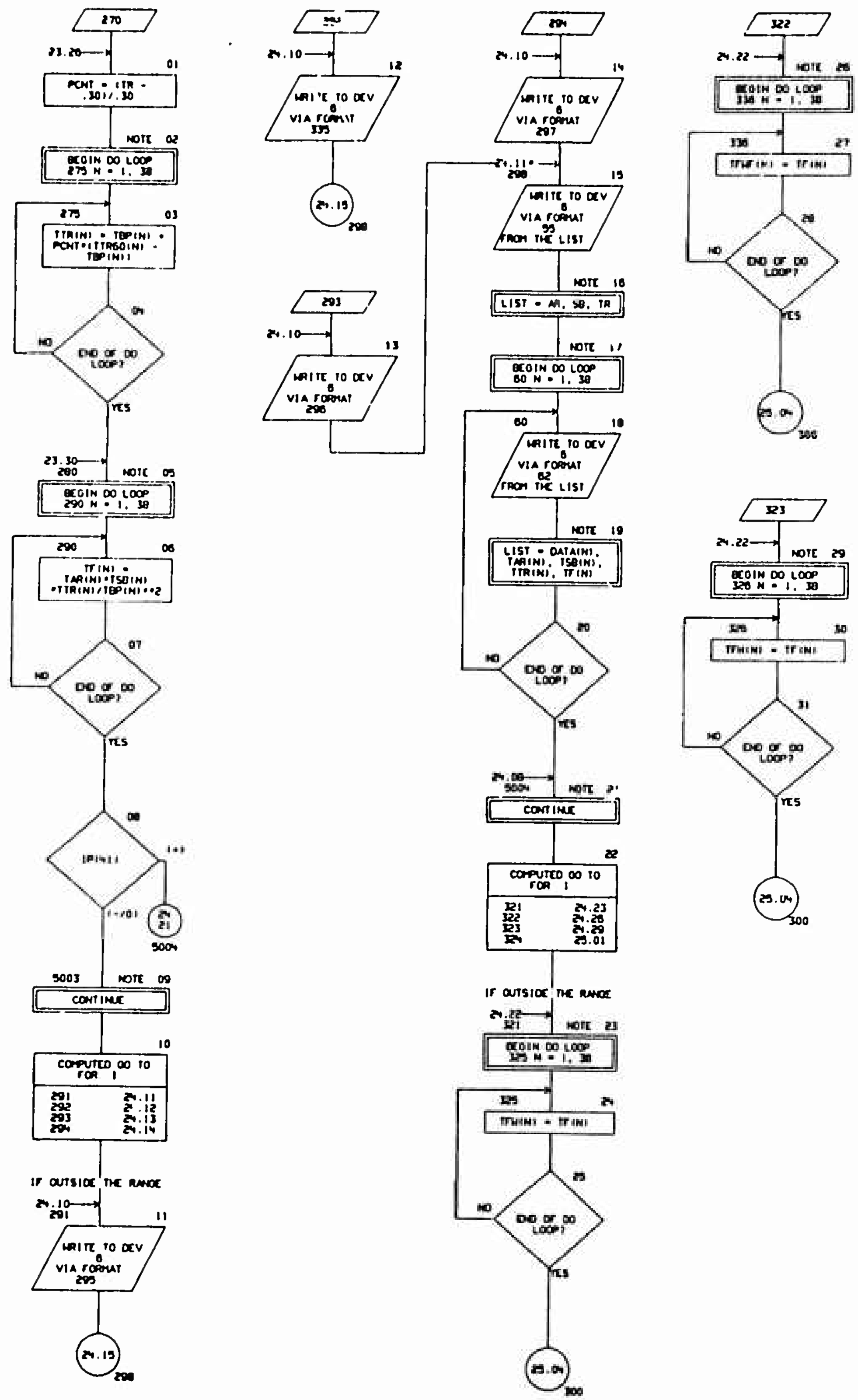


CHART TITLE - SUBROUTINE SVFTAB

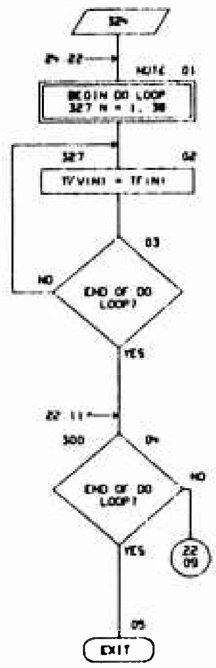


CHART TITLE - NON-PROCEDURAL STATEMENTS

```
          COMMON    SVF(180),DATA(312),TFW(38),TFH(38),TFV(38),TFWF(38)
          COMMON /IPRINT/IP(80)
          COMMON /MISC/ XMISC(100)
          DIMENSION TABLE(38),TBP(38),TAR2(38),TAR6(38),
         1 TS80(38),TS860(38),TTR0(38),TTR60(38)
          DIMENSION TAR(38),TSB(38),TTR(38),TF(38)
          EQUIVALENCE (TABLE(1),DATA(  1)),(  TBP(1),DATA( 39)),
                      ( TAR2(1),DATA( 77)),( TAR6(1),DATA(115)),
                      ( TS80(1),DATA(153)),(TS860(1),DATA(191)),
                      ( TTR0(1),DATA(229)),(TTR60(1),DATA(267))
295       FORMAT(1H1,40X,27H*** WING (FIXED OR AFT) ***,
         221X,21H** SVFTAB - IP(4)) **//)
335       FORMAT(1H1,40X,22H*** WING (FORWARD) ***,
           26X,21H** SVFTAB - IP(4)) **//)
296       FORMAT(1H1,35X,23H*** HORIZONTAL TAIL ***,
           30X,21H** SVFTAB - IP(4)) **//)
297       FORMAT(1H1,35X,21H*** VERTICAL TAIL ***,
           32X,21H** SVFTAB - IP(4)) **//)
55        FORMAT(1H ,35X,32HFLUTTER PARAMETER VS MACH NUMBER//
                      18X,8HMACH NO.,12X,5HAR = ,F5.2,4X,13HSWEEP(C/4) = ,
           F4.1,4H DEG,3X,8HTAPER = ,F4.3,10X,9HCOMPOSITE//)
62        FORMAT(17X,F5.3,18X,F5.4,15X,F5.4,15X,F5.4,15X,F5.4)
```

```
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
                          FUNCTION TBL
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''

   NOTE-  THIS FUNCTION SUBPROGRAM IS DESCRIBED IN REPORT NA-63-1473
   NOTE-  THIS VERSION OF THE FUNCTION IS IN FORTRAN IV


          PROGRAM TO COMPUTE BOUNDARY LAYER TEMPERATURE

          C J MAC MILLER                  12 MARCH,1964

   THE EQUATION IS PROGRAMMED AS FOLLOWS

     BOUNDARY LAYER TEMPERATURE =

          TBL(XMACH,PO,TO,TSK,DIST,XLAMDA,REY,METHOD,IER,TOTI)

     WHERE - XMACH = LOCAL MACH NUMBER  DIMENSIONLESS
                PO = LOCAL PRESSURE, PSF
                TO = LOCAL TEMPERATURE, DEG RANKINE
               TSK = SKIN TEMPERATURE, DEG RANKINE
              DIST = CHARACTERISTIC LENGTH, DISTANCE AFT OF LEADING
                     EDGE, FEET
            XLAMDA = ANGLE OF SWEEP,DEGREES
               REY = TRANSITION REYNOLDS NUMBER, DIMENSIONLESS
            METHOD = INDICATOR TO CONTROL TYPE OF SOLUTION
             IF METHOD = 0 OR 1, FUNCTION RETURNS SKIN TBL
                METHOD = 2, FUNCTION RETURNS STAGNATION TBL
               IER = ERROR INDICATOR
                 IF IER = 0, NO ERROR, SOLUTION CRITERIA WAS MET
                    IER = 2, SOLUTION FOR BOUNDARY LAYER TEMP DID
                             NOT CONVERGE IN 100 PASSES, VALUE RE-
                             TURNED IS BASED ON RECOV OF  875
              TOTI = TOTAL TEMPERATURE

   THE VALUE OF BOUNDARY LAYER TEMPERATURE IS RETURNED IN DEGREES
   RANKINE TO THE CALLING PROGRAM



   VARIATION OF SPECIFIC HEAT, VISCOSITY, CONDUCTIVITY, AND DENSITY
   ARE TAKEN INTO ACCOUNT
   REFERENCE - C(P) - NACA TM NO. L8J14, 10 DEC, 1948
               VISCOSITY - WADC TR55-169
               CONDUCTIVITY - U.S  STANDARD ATM, 1962
               DENSITY - PERFECT GAS
   IF NO VALUE OF REY IS SUPPLIED, THE FUNCTION WILL USE ONE MILLION
```

CHART TITLE - FUNCTION TBL(XMACH,PO,TO,TSK,DIST,XLAMDA,REY,METHOD,IER,TOT1)

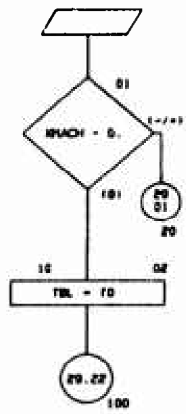


CHART TITLE - FUNCTION TBL(XMACH,PO,TO,TSK,DIST,XLAMDA,REY,METHOD,ICR,TOT1)

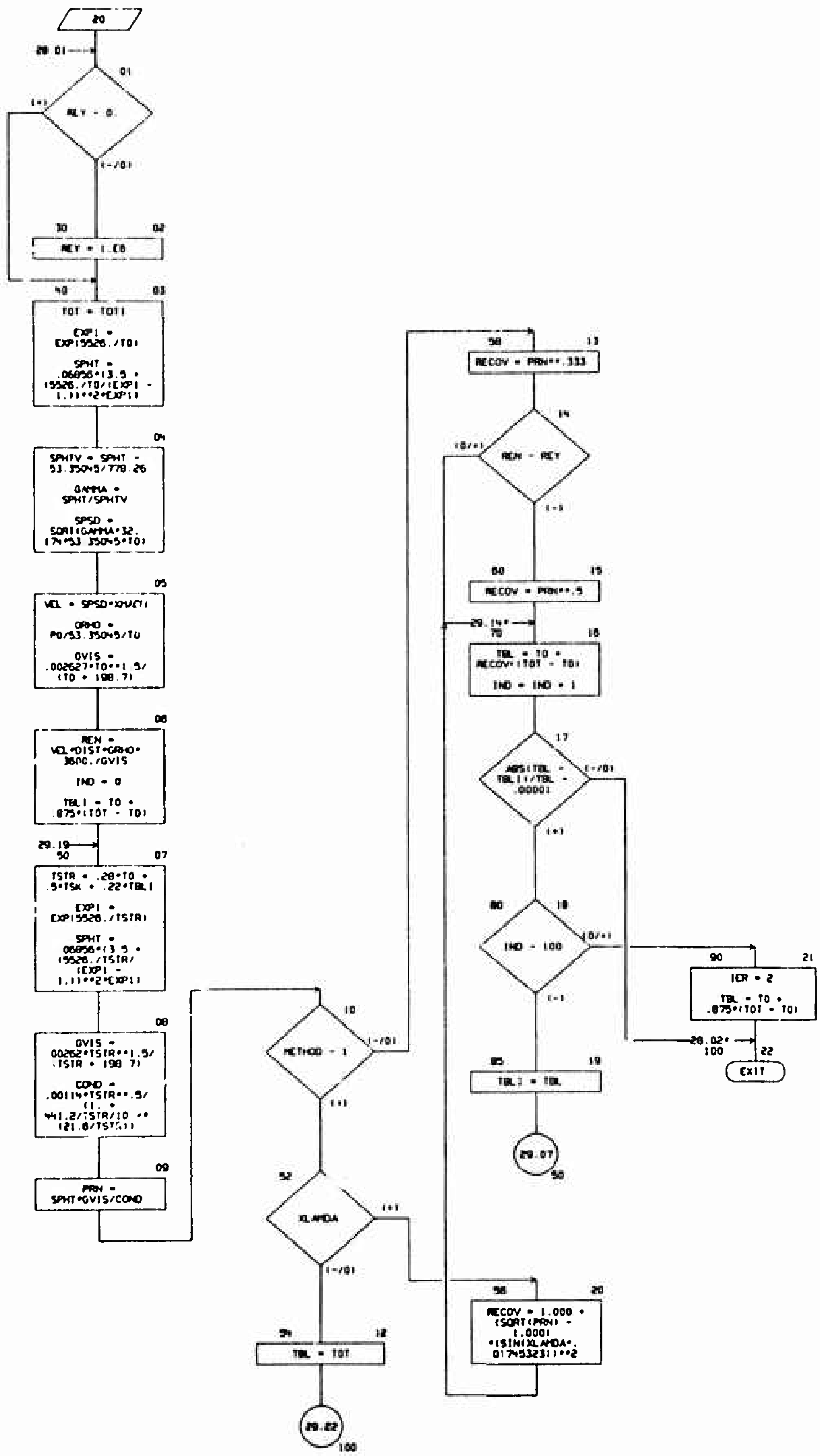


CHART TITLE - INTRODUCTORY COMMENTS

FUNCTION TEMALT

CHART TITLE - FUNCTION TEMALT(ALT,IT)

TEMALT IS A FUNCTION SUBPROGRAM FOR COMPUTING AIR TEMPERATURE AS A FUNCTION OF PRESSURE ALTITUDE FOR FOR VARIOUS MODEL ATMOSPHERES.
TWO PARAMETERS ARE USED IN THE CALLING STATEMENT -
(1) THE PRESSURE ALTITUDE AS WOULD BE INDICATED BY AN ALTIMETER CALIBRATED IN GEOPOTENTIAL ALTITUDE FOR U.S STD ATMOSPHERE,1962.
(2) AN INDICATOR OF THE MODEL ATMOSPHERE UNDER CONSIDERATION
USE 1.0 FOR THE U.S STANDARD ATMOSPHERE, 1962
USE 2.0 FOR THE COLD ATMOSPHERE PER TABLE II OF MIL-STD-210A
USE 3.0 FOR THE HOT ATMOSPHERE PER TABLE III OF MIL-STD-210A.
USE 4.0 FOR POLAR ATMOSPHERE PER TABLE IV OF MIL-STD-210A
USE 5.0 FOR TROPICAL ATMOSPHERE PER TABLE V OF MIL-STD-210A.
NOTE- THESE INDICATORS CAN BE EITHER NEGATIVE OR POSITIVE.

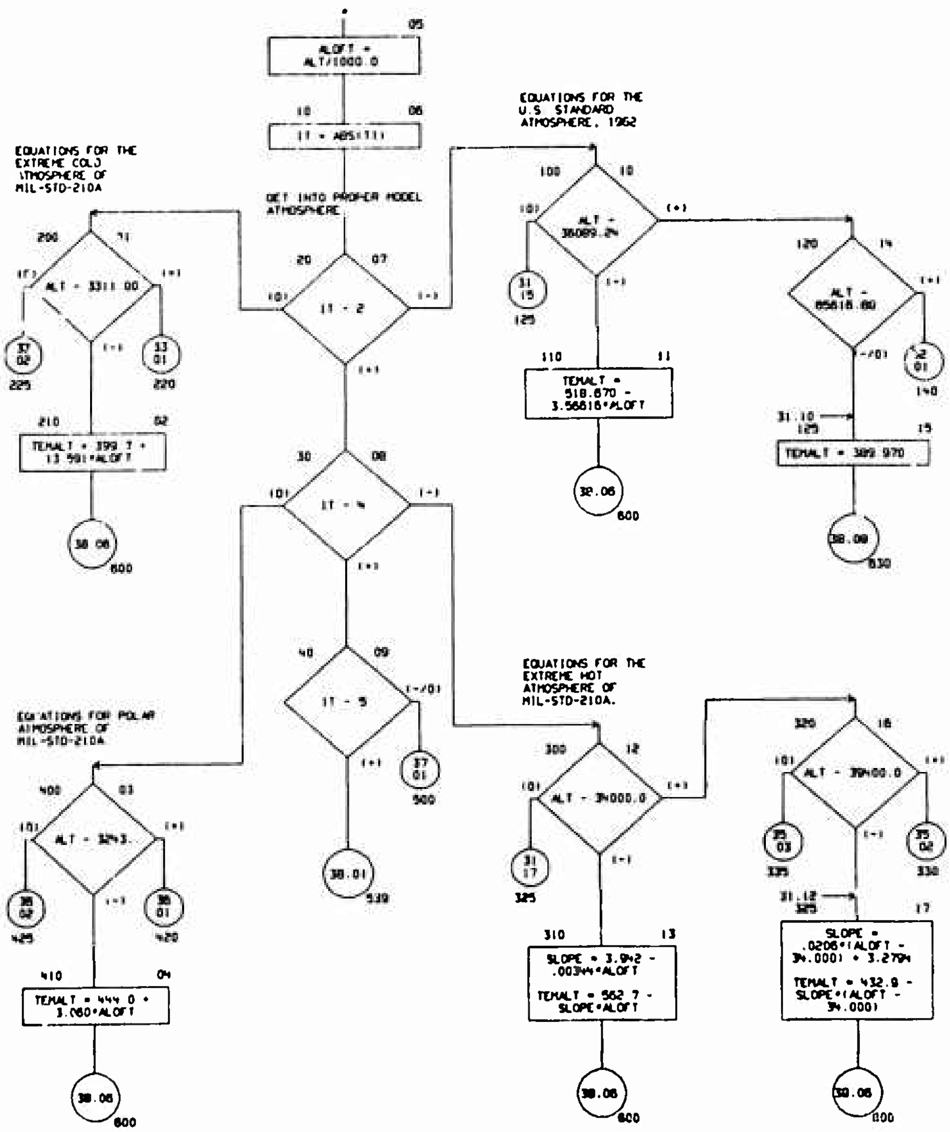


CHART TITLE - FUNCTION TEMALT(ALT,T1)

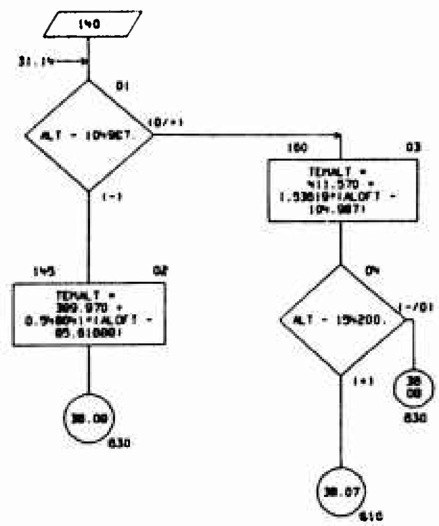
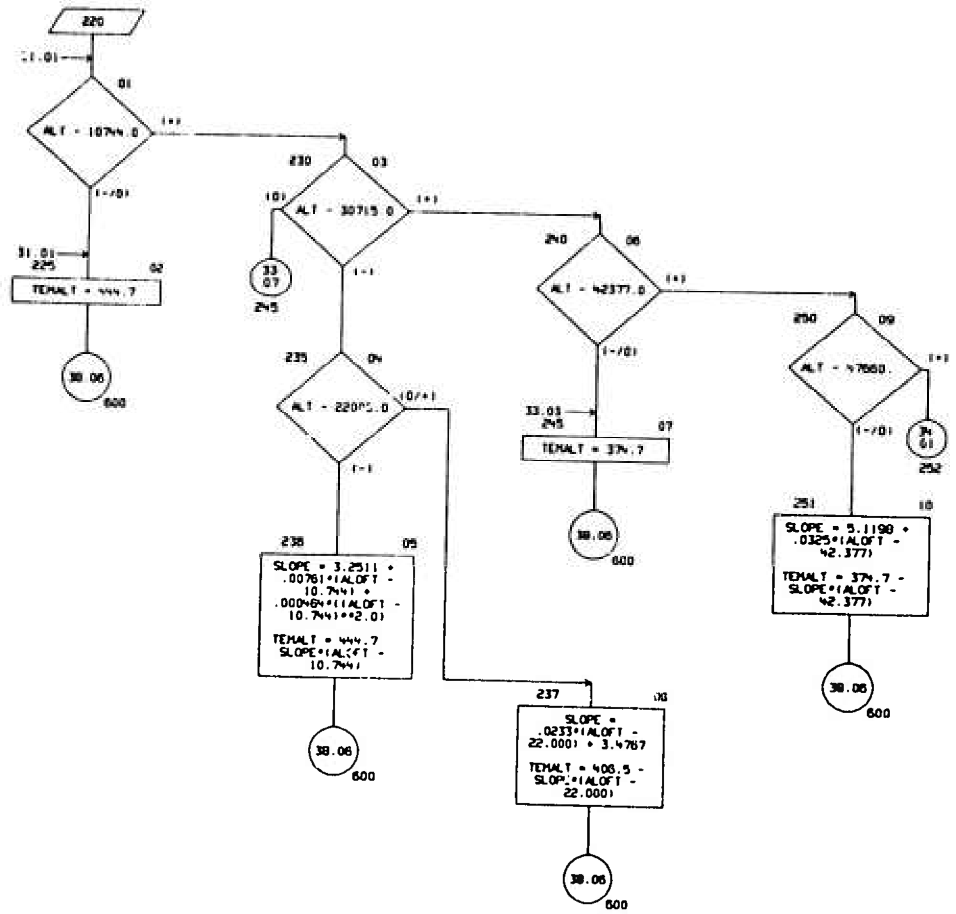


CHART TITLE - FUNCTION TEMALT(ALT,T1)



AUTOFLOW CHART SET - SWEEP FLUTTER AND TEMPERATURE PAGE 34

11/07/73

CHART TITLE - FUNCTION TEMALT(ALT,T1)

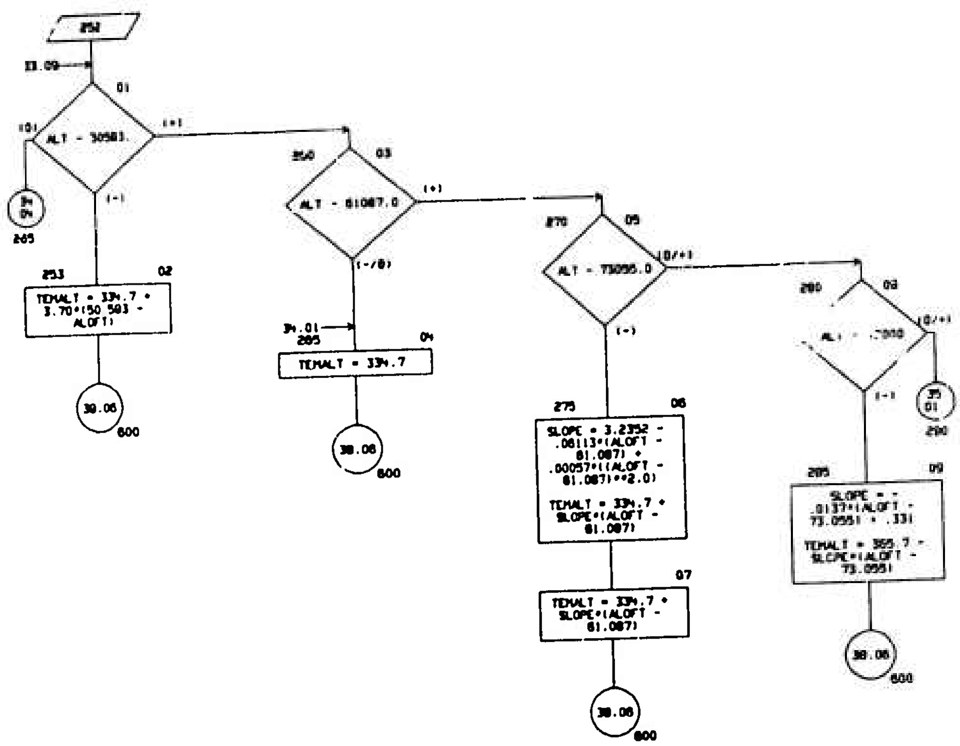


CHART TITLE - FUNCTION TEMALT(ALT,TI)

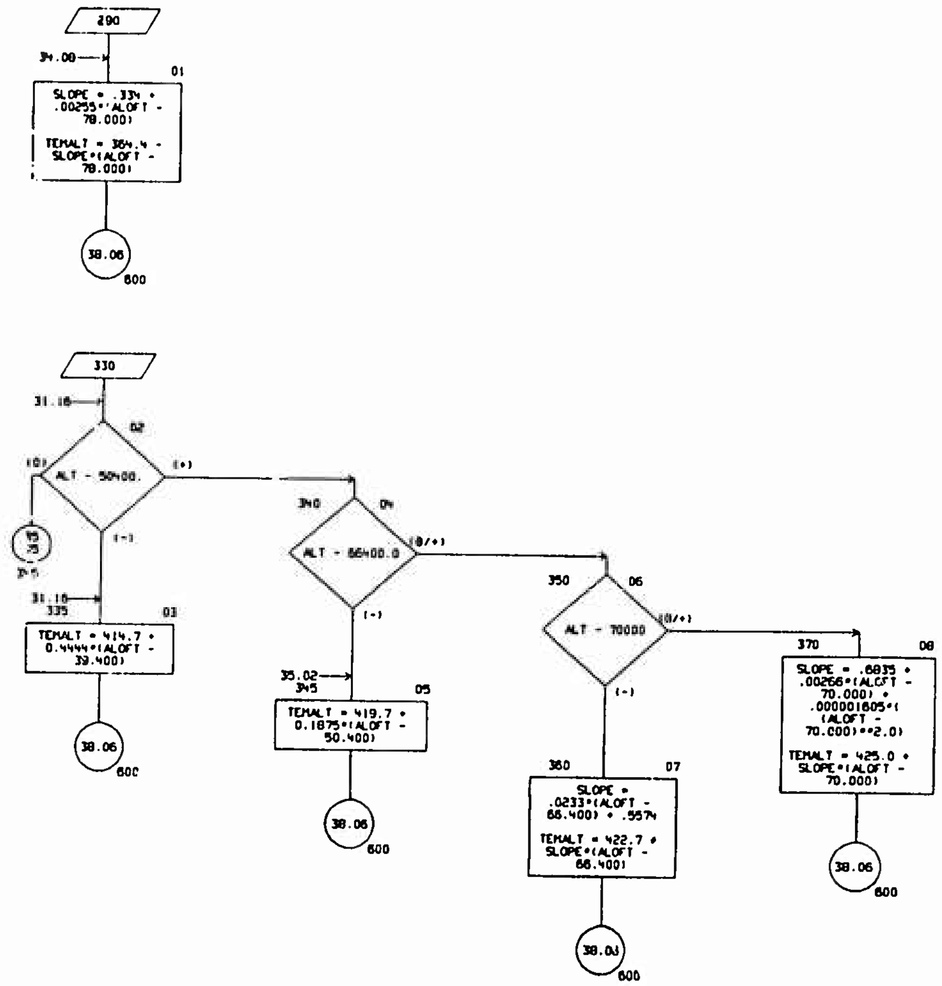


CHART TITLE - FUNCTION TEMALT(ALT,TI)

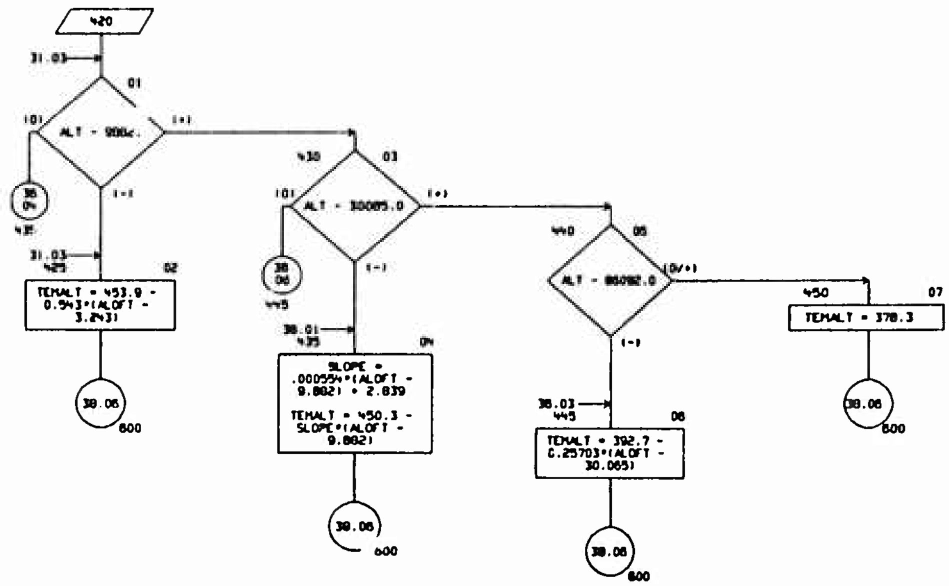


CHART TITLE - FUNCTION TEMALT(ALT,T)

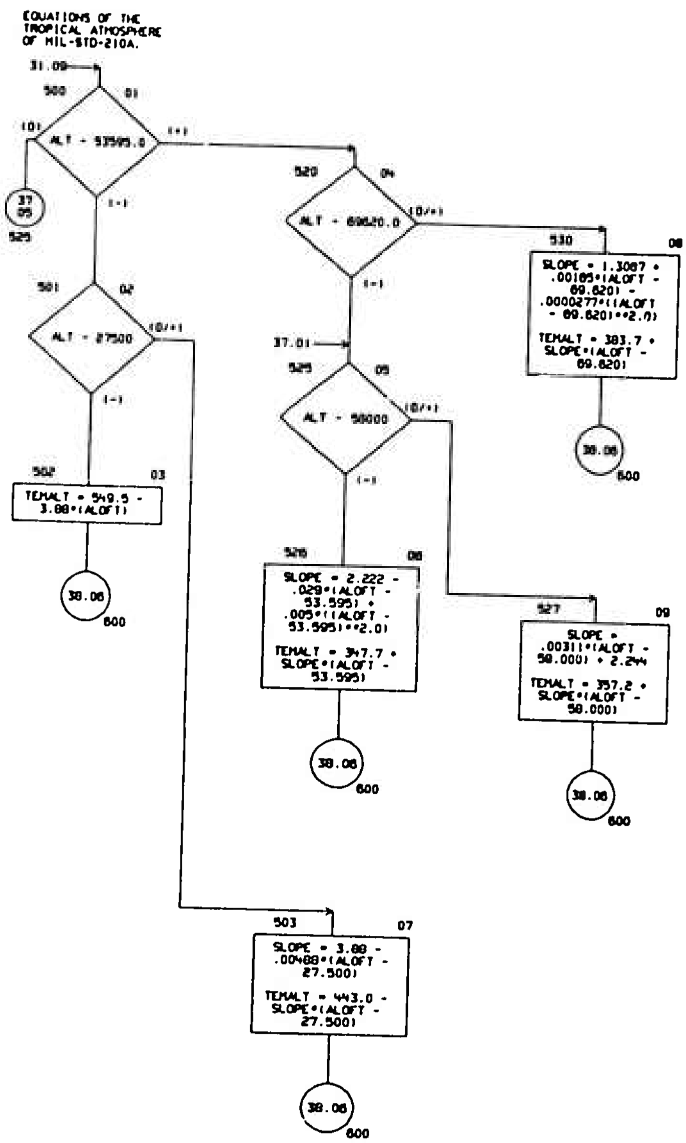


CHART TITLE - FUNCTION TEMALT(ALT,TI)

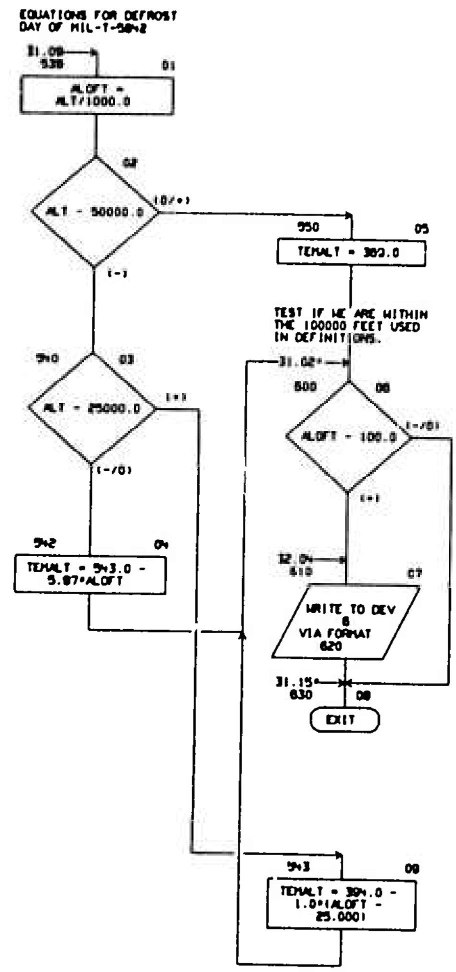


CHART TITLE - NON-PROCEDURAL STATEMENTS

```
620     FORMAT(1H 0X50HWARNING- ALTITUDE IS BEYOND VALID RANGE OF TEMALT.)
```

CHART TITLE - INTRODUCTORY COMMENTS

```
))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))
                       SUBROUTINE TEMPER
((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((
```

CHART TITLE - SUBROUTINE TEMPER(XMACH,ALT,PRESSI,TLOC,TTOT,SUN,XSKIN,XSKINF,IER

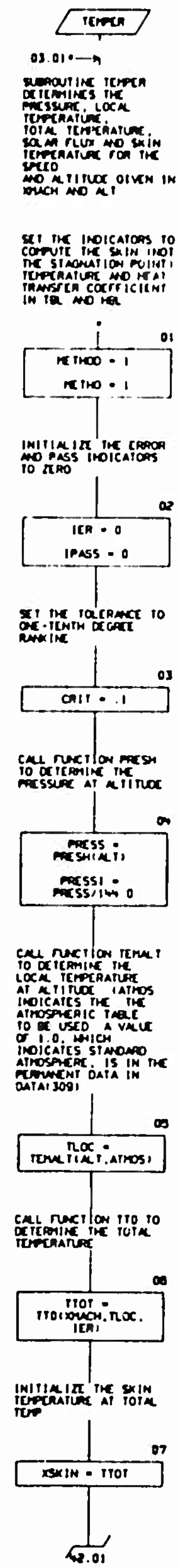
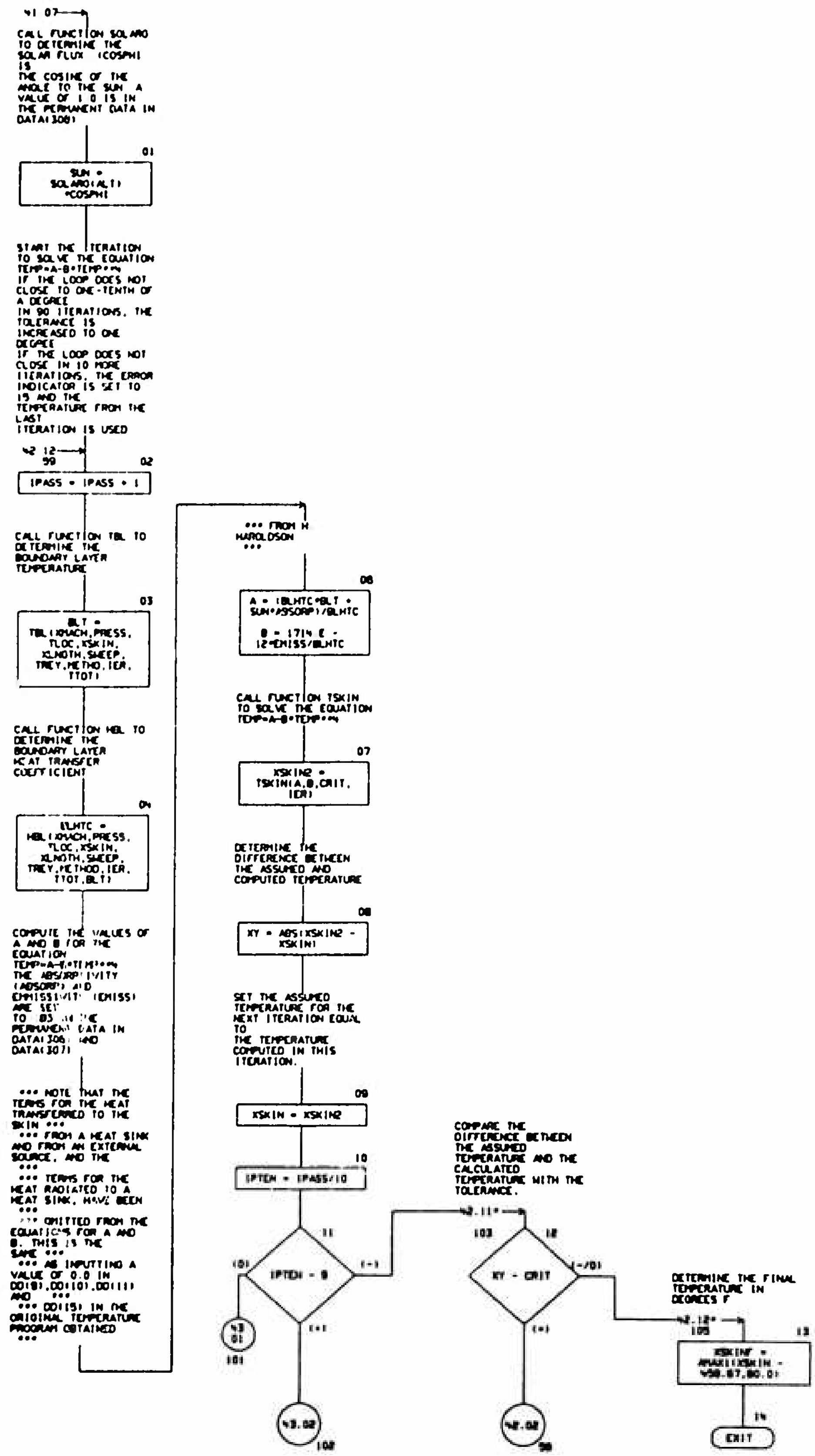


CHART TITLE - SUBROUTINE TEMPER(XMACH,ALT,PRESS,TLOC,TTOT,SUN,XSKIN,XSKINF,IER



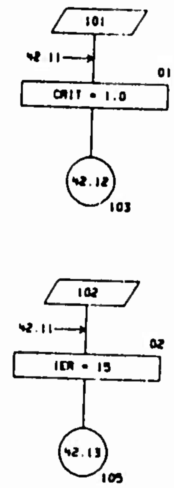

101
42.11
01
CRIT = 1.0
42.12
103
102
42.11
02
IER = 15
42.13
105

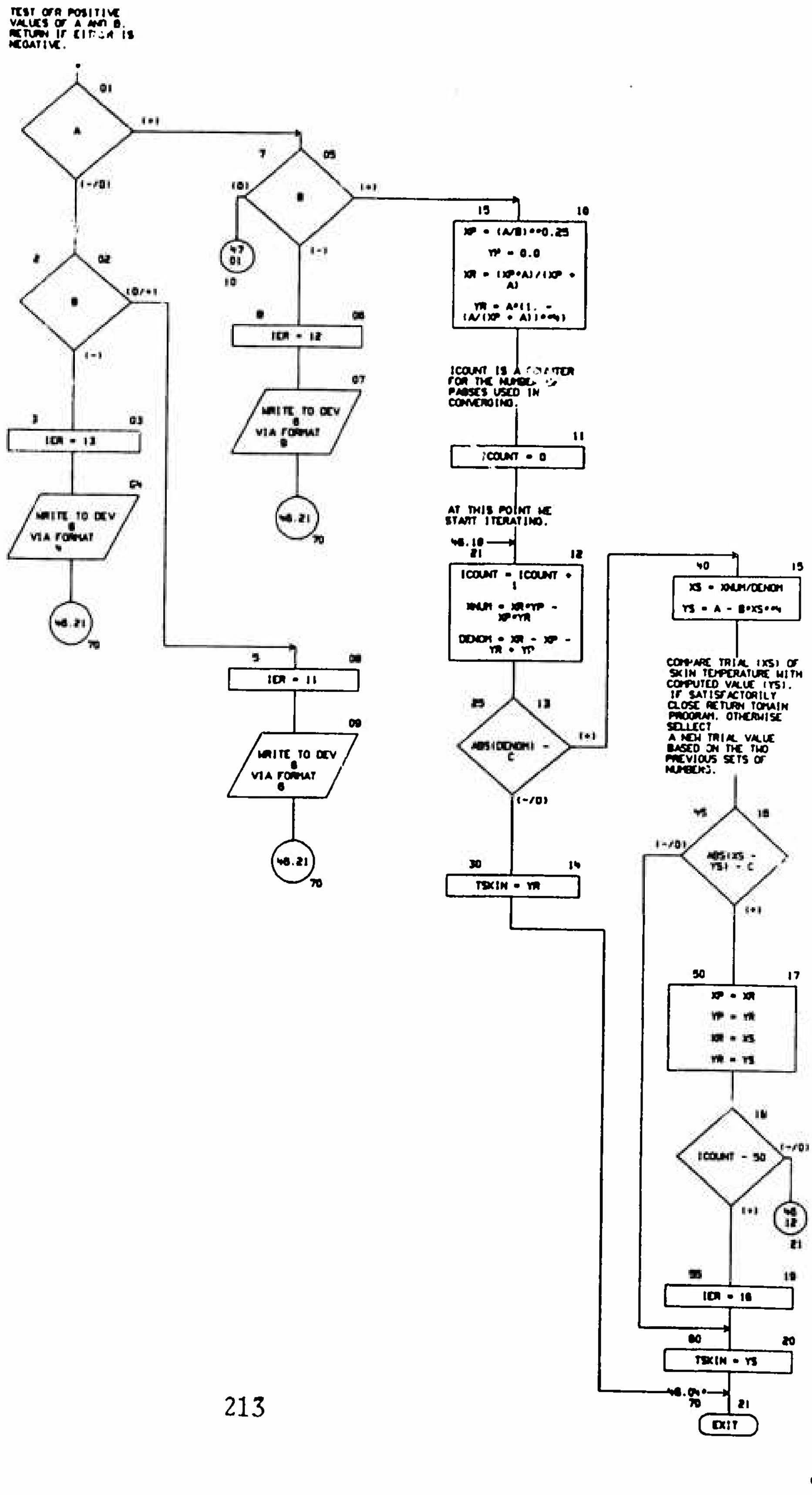
CHART TITLE - NON-PROCEDURAL STATEMENTS

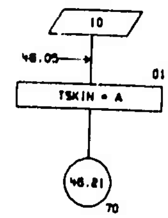
```
COMMON     SVF(180),DATA(312),TFM(30),TFH(30),TFV(30),TFWF(30)
EQUIVALENCE        (EMISS ,DATA(306)),(ABSORP,DATA(307)),
 (COSPHI,DATA(308)),(ATMOS ,DATA(309)),(XLNGTH,DATA(310)),
 (TREY  ,DATA(311)),(SWEEP ,DATA(312))
```

CHART TITLE - INTRODUCTORY COMMENTS

```
))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))
                              FUNCTION TSKIN
((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((

     TSKIN IS A FUNCTION SUBPROGRAM FOR SOLVING TEMPERATURES FROM
  AN EQUATION OF FIRST AND FOURTH POWERS IN THE FORM
     T = A-B* T**4
  ALL TEMPERATURES ARE IN DEGREES RANKINE. THE ROOT IS REAL AND
  POSITIVE
      THE FUNCTION IS CALLED BY THE STATEMENT
      TEMP = TSKIN(A,B,C,IER)
  WHEN A AND B ARE CONSTANTS, C IS A CRITERIAN FOR SATISFACTORY
  CONVERGENCE, AND IER IS AN ERROR INDICATOR
```

CHART TITLE - FUNCTION TSKIN(A,B,C,IER)



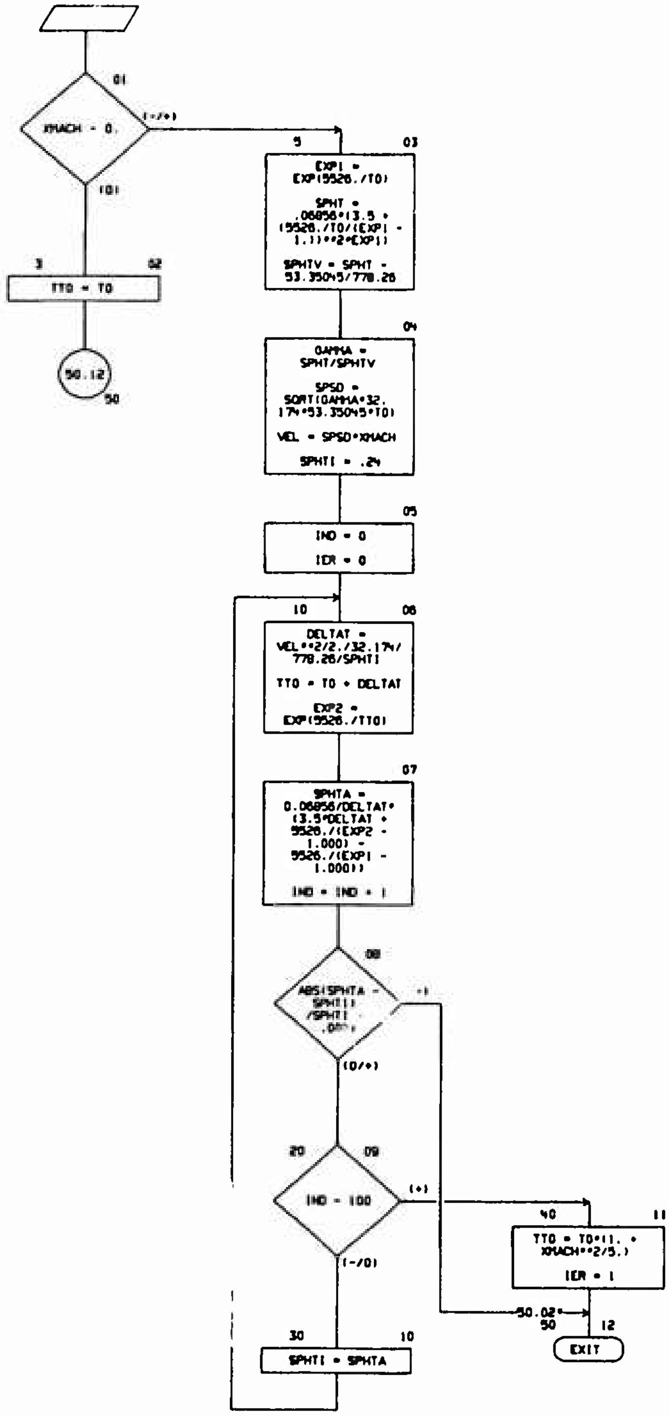

10
48.05
01
TSKIN = A
48.21
70

```
7       FORMAT(1H0 10X,'BOTH A AND B ARE NEGATIVE. TSKIN IS DISCONTINUED')
8       FORMAT(1H0 10X,'A IS NEGATIVE OR ZERO. TSKIN IS DISCONTINUED.')
9       FORMAT(1H0 10X,'B IS NEGATIVE. TSKIN IS DISCONTINUED')
```

CHART TITLE - INTRODUCTORY COMMENTS

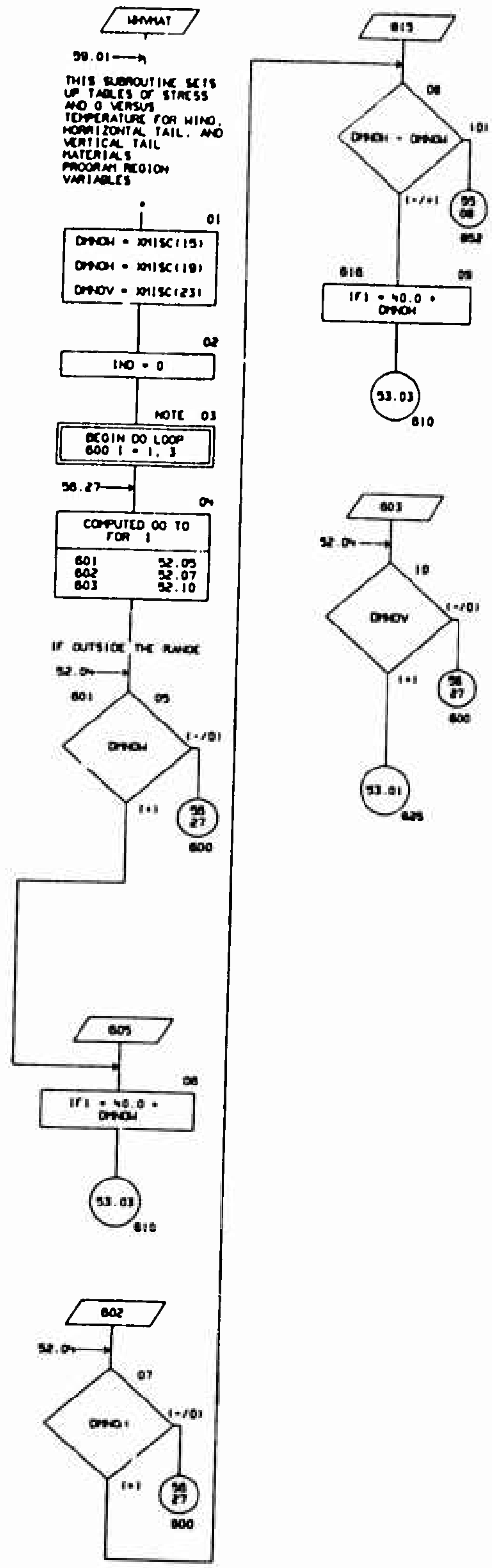
```
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
                          FUNCTION TTO
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''

   TOT4,  MACMILLER,  DEPT 381,  GROUP 132,  STATION 2-17,
   NOTE-  THIS FUNCTION SUBPROGRAM IS DESCRIBED IN REPORT NA-63-1473.
   NOTE-  THIS VERSION OF THE FUNCTION IS IN FORTRAN IV.


              PROGRAM TO COMPUTE TOTAL TEMPERATURE
               C.J.MAC MILLER    10 MARCH 1964

   THE EQUATION IS PROGRAMMED AS....

      TOTAL TEMPERATURE = TTO(XMACH,TO,IER)

       WHERE - XMACH = FREESTREAM MACH NUMBER - DIMENSIONLESS
                  TO = FREESTREAM STATIC TEMPERATURE - DEG RANKINE
                 IER = ERROR INDICATOR
                  IF IER = 0, NO ERROR, SOLUTION CRITERIA WAS MET
                     IER = 1,CONVERGENCE WAS NOT ACHIEVED WITHIN 100
                  PASSES, VALUE OF TTO RETURNED IS BASED ON CONSTANT
                  GAMMA

   THE BASIC DERIVATION RELATES CHANGE IN ENTHALPY OF AIR TO CHANGE
   IN KINETIC ENERGY

   VARIATION OF SPECIFIC HEAT WITH TEMPERATURE IS TAKEN INTO ACCOUNT
   REFERENCE - NACA TN NO. L8J14, 10DEC,1948
```

CHART TITLE - FUNCTION TTO(XMACH,TO,IER)



SUBROUTINE MHVMAT

CHART TITLE - SUBROUTINE WHVMAT



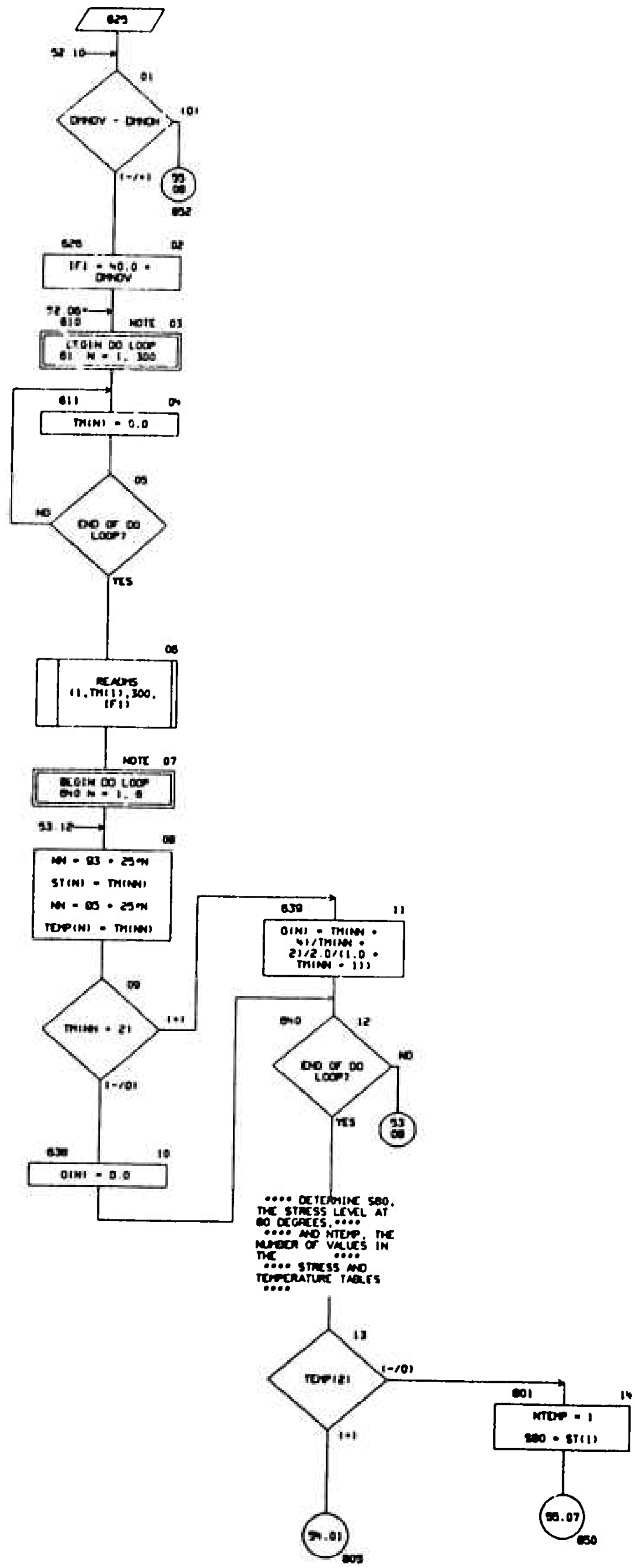
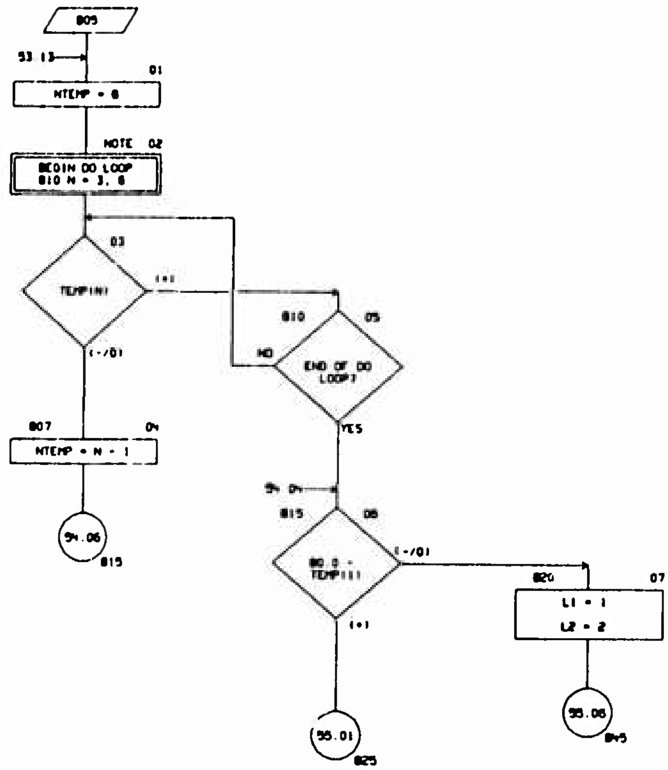


CHART TITLE - SUBROUTINE WHVHAT



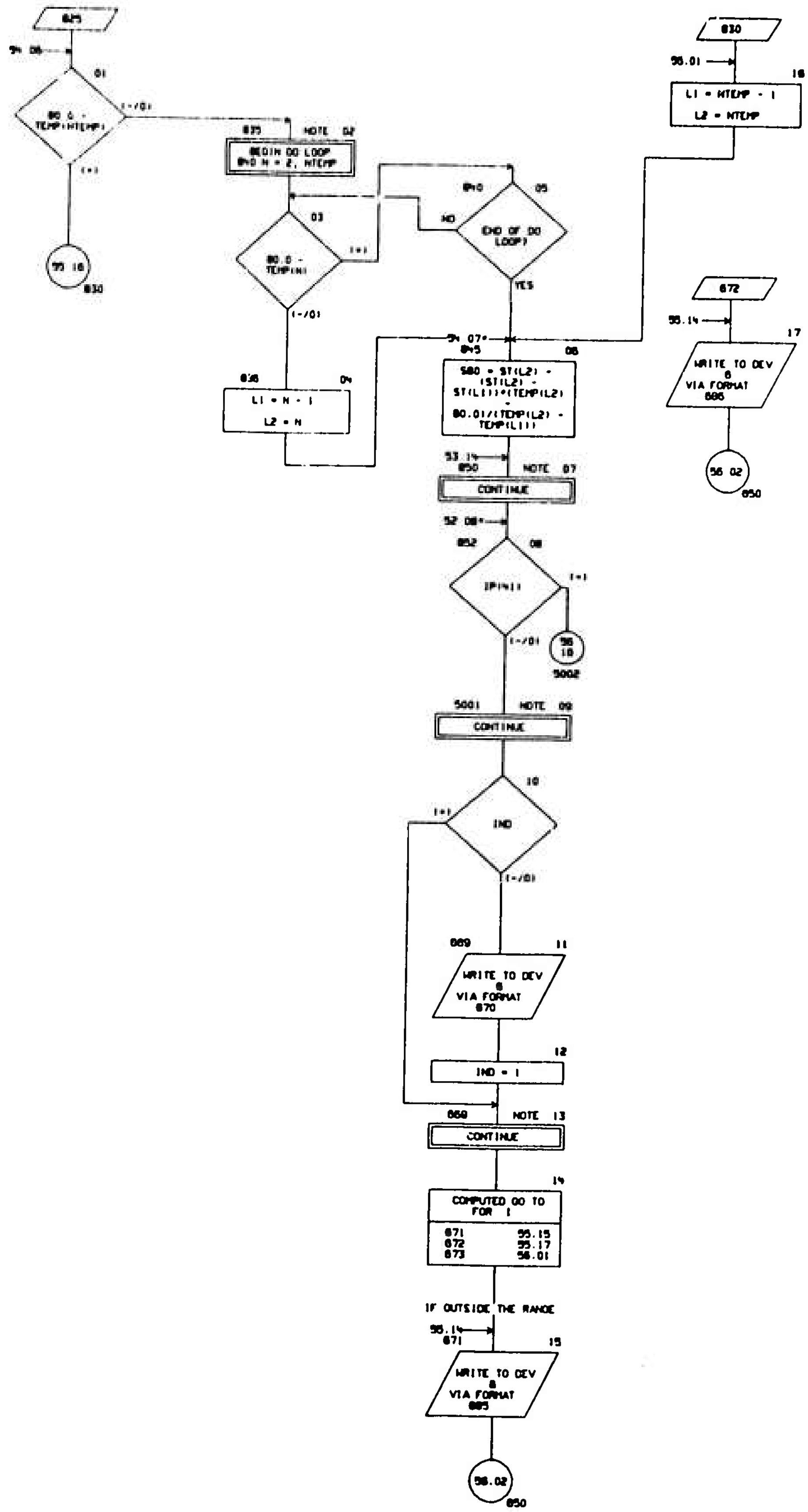

830
55.01
16
L1 = NTEMP - 1
L2 = NTEMP
01
80.0 - TEMP(NTEMP)
(-/0)
(+)
835
NOTE 02
BEGIN DO LOOP
840 N = 2, NTEMP
840
05
NO
END OF DO LOOP?
YES
03
80.0 - TEMP(N)
(+)
(-/0)
836
04
L1 = N - 1
L2 = N
845
06
580 = ST(L2) - (ST(L2) - ST(L1))*(TEMP(L2) - 80.0)/(TEMP(L2) - TEMP(L1))
672
55.14
17
WRITE TO DEV 6 VIA FORMAT 686
56 02
850
53.14
850
NOTE 07
CONTINUE
52 08
852
08
IP(N1)
(+)
(-/0)
5002
5001
NOTE 09
CONTINUE
10
IND
(+)
(-/0)
869
11
WRITE TO DEV 6 VIA FORMAT 870
12
IND = 1
868
NOTE 13
CONTINUE
14
COMPUTED GO TO FOR I
671 55.15
672 55.17
673 56.01
IF OUTSIDE THE RANGE
55.14
671
15
WRITE TO DEV 6 VIA FORMAT 685
56.02
850

CHART TITLE - SUBROUTINE WHVMAT

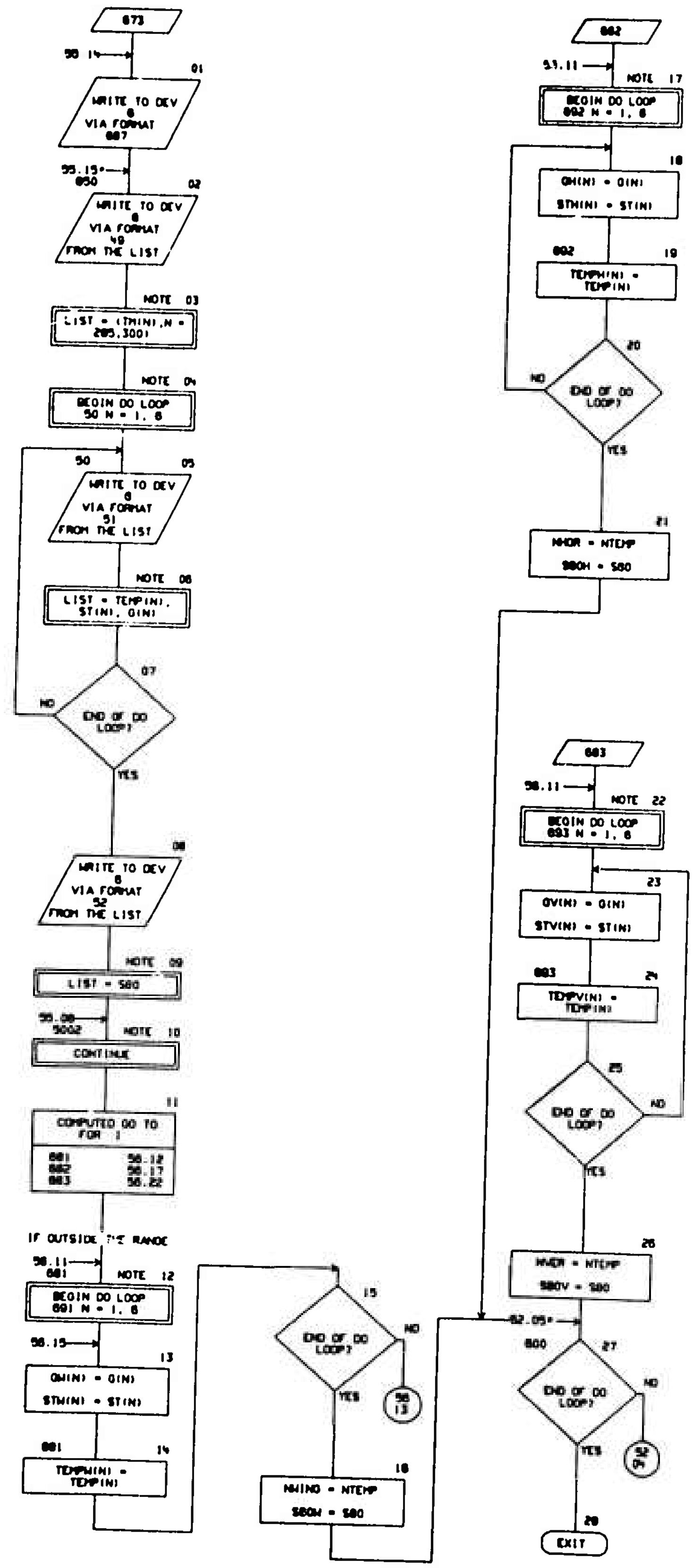


CHART TITLE - NON-PROCEDURAL STATEMENTS

```
          COMMON     SVF(180),DWI(3(2),TFW(30),TFH(30),TFV(30),TFWF(30),
                     TF(30),XMNI(20),Q(20),QQ(20),XMATOP(20),
                     OW(6),OH(6),OV(6)
          COMMON /IPRINT/IP(80)
          COMMON /MISC/XMISC(100)
          DIMENSION  STW(6),STH(6),STV(6),TEMPW(6),TEMPH(6),TEMPV(6)
          DIMENSION TH(300),ST(6),TEMP(6),O(6)
          EQUIVALENCE                              (SVF(139),STW   (1)),
           (SVF(145),TEMPW (1)),(SVF(153),STH   (1)),(SVF(159),TEMPH (1)),
           (SVF(167),STV   (1)),(SVF(173),TEMPV (1)),(SVF(151),NWING    ),
           (SVF(165),NHOR     ),(SVF(179),NVER     ),(SVF(152),S80W     ),
           (SVF(166),S80H     ),(SVF(180),S80V     )
870       FORMAT(1H1,88X,21H** WHVMAT - IP(41) **)
885       FORMAT(   45X,12H*** WING *** )
886       FORMAT(  /40X,23H*** HORIZONTAL TAIL *** )
887       FORMAT(  /40X,21H*** VERTICAL TAIL *** )
49        FORMAT(10X,8A10/10X,8A10//30X,11HTEMPERATURE,5X,12HSTRESS (PSI),
            10X,7HQ (PSI)/)
51        FORMAT(33X,F6.0,9X,F8.0,9X,F11.0)
52        FORMAT(/30X,20HSTRESS AT 80 DEGREES,F10.0)
```

CHART TITLE - INTRODUCTORY COMMENTS

```
))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))
                          SUBROUTINE WHVQQ
((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((
```

CHART TITLE - SUBROUTINE WHVQO

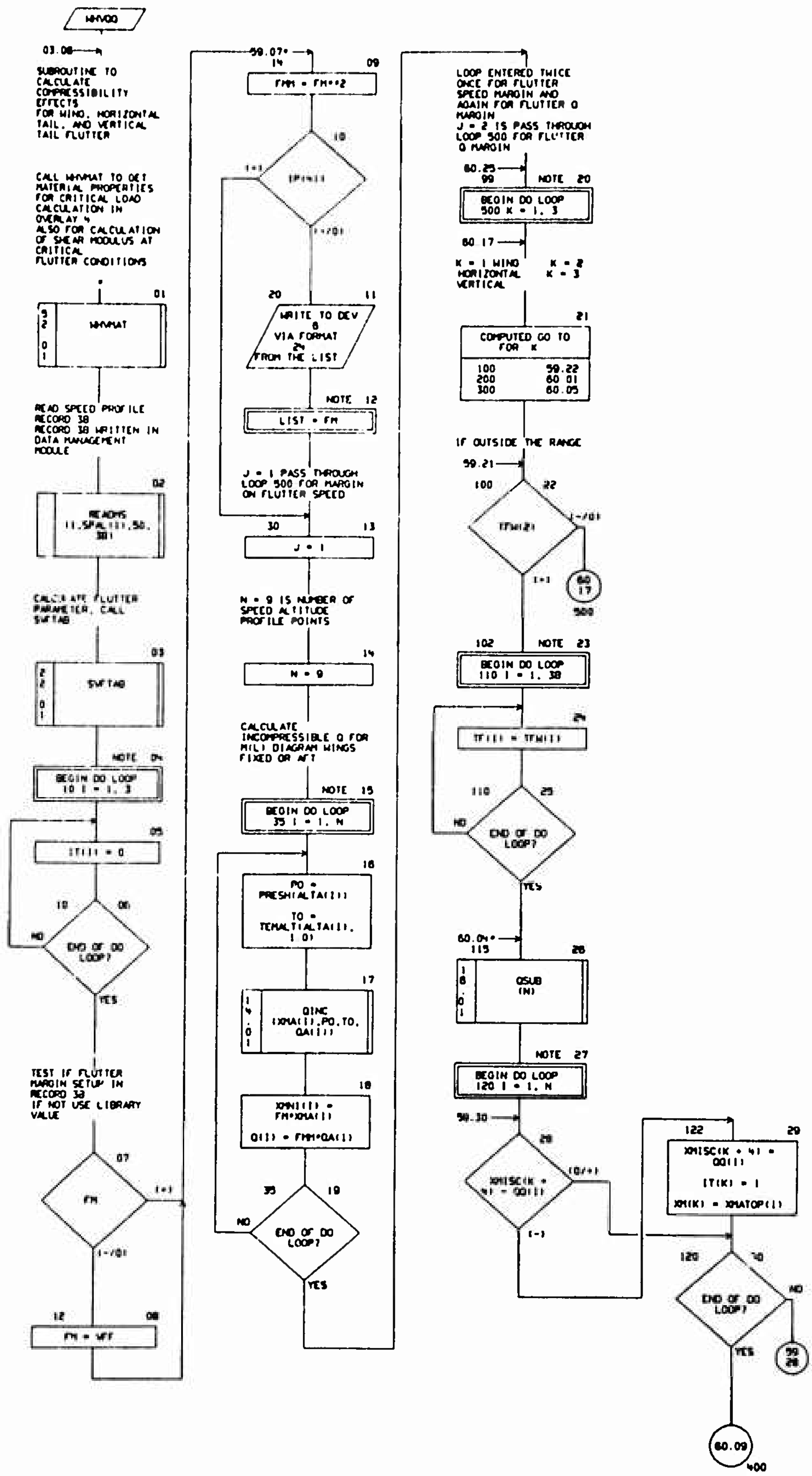


CHART TITLE - SUBROUTINE WHVQQ

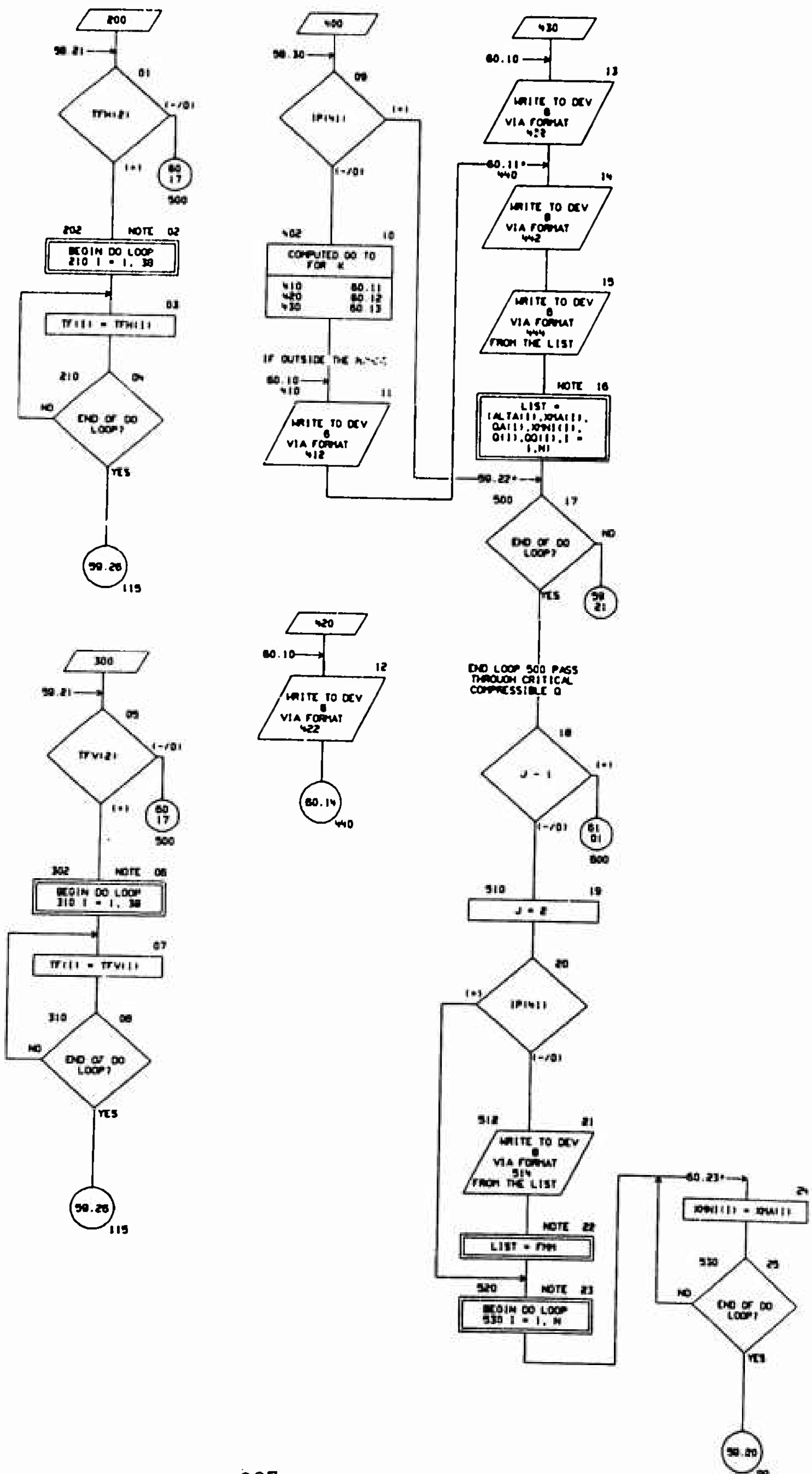


CHART TITLE - SUBROUTINE WHVOO

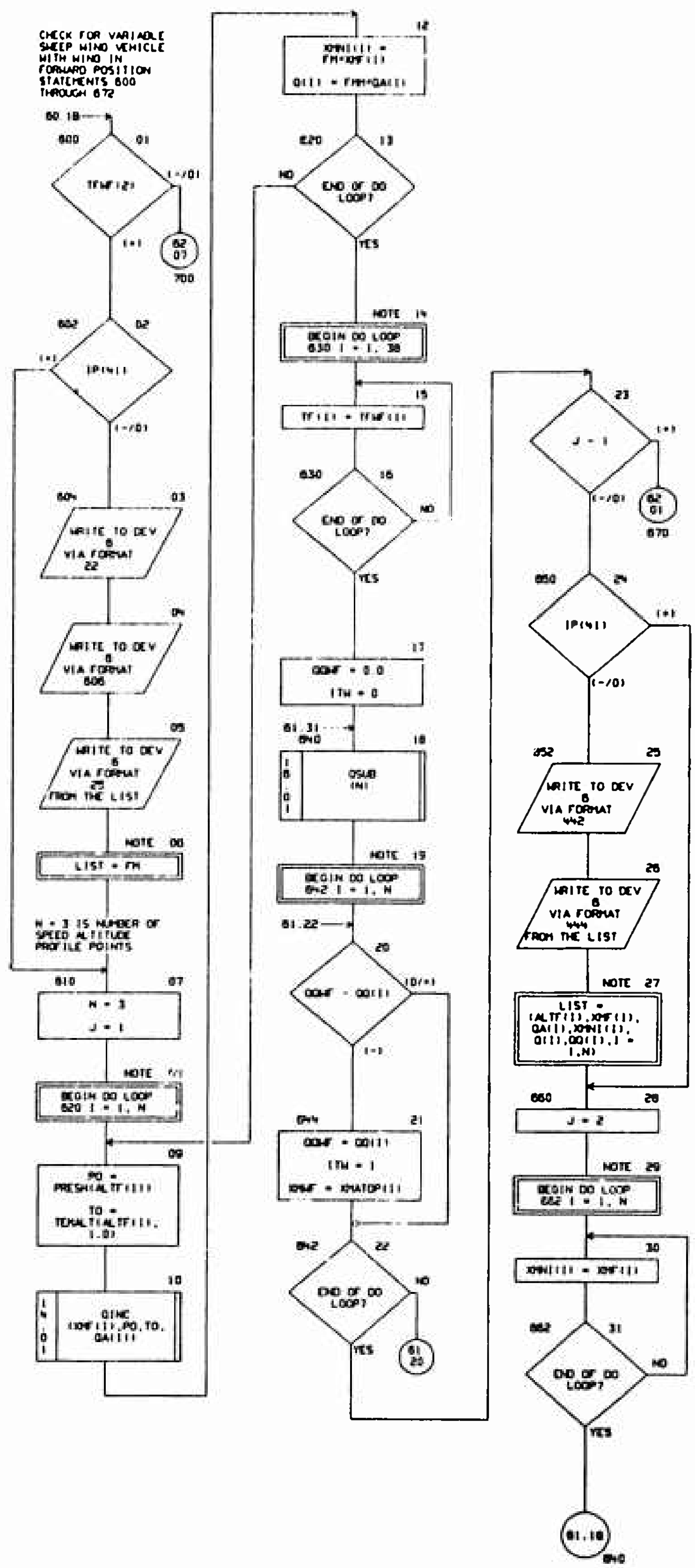


CHART TITLE - SUBROUTINE HHVOO

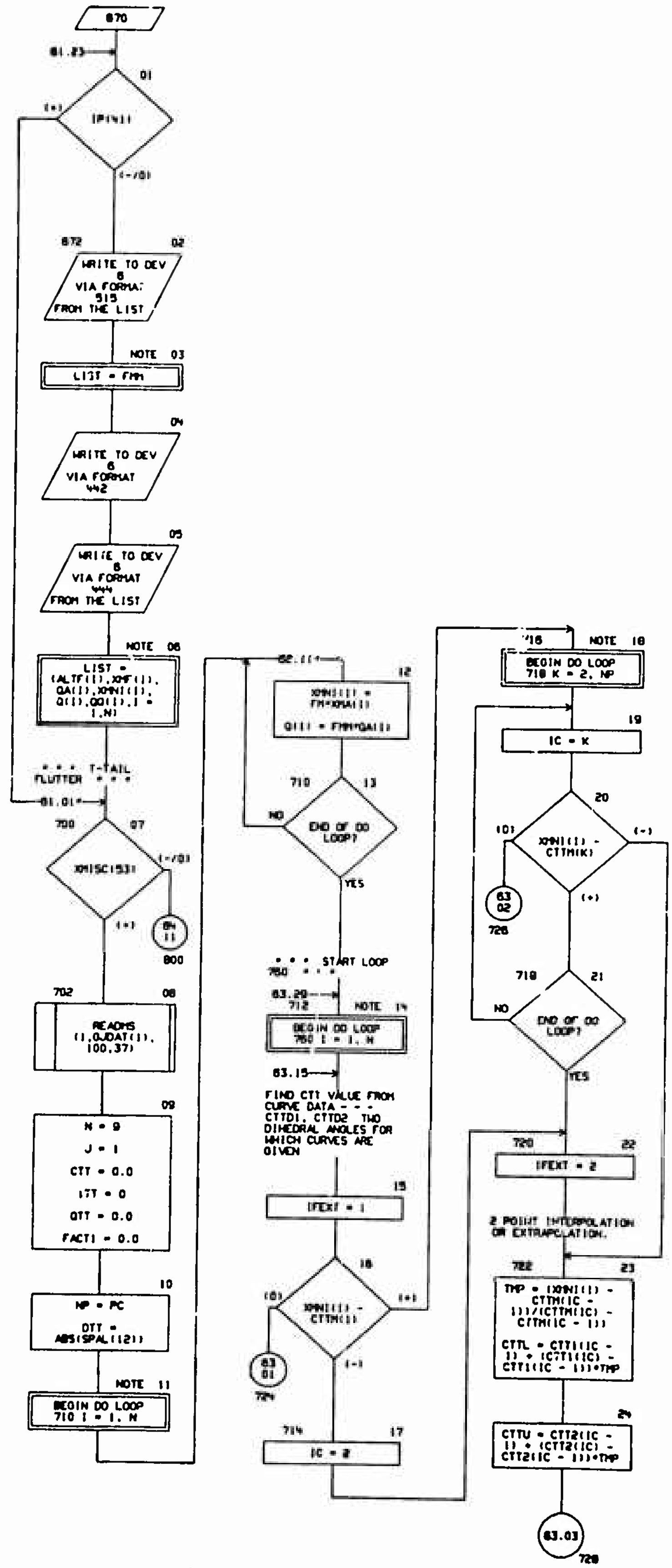


CHART TITLE - SUBROUTINE HHVOO

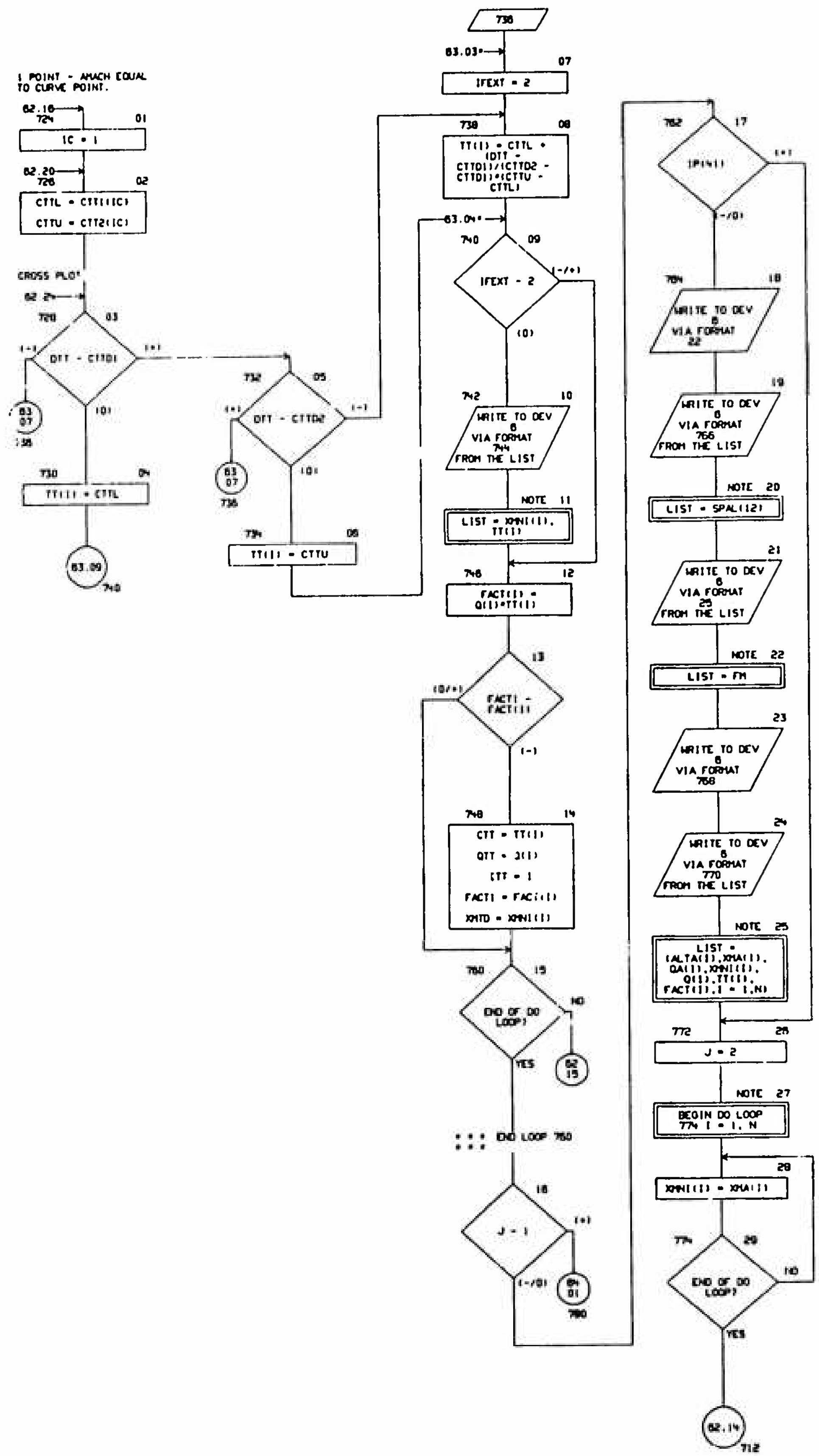
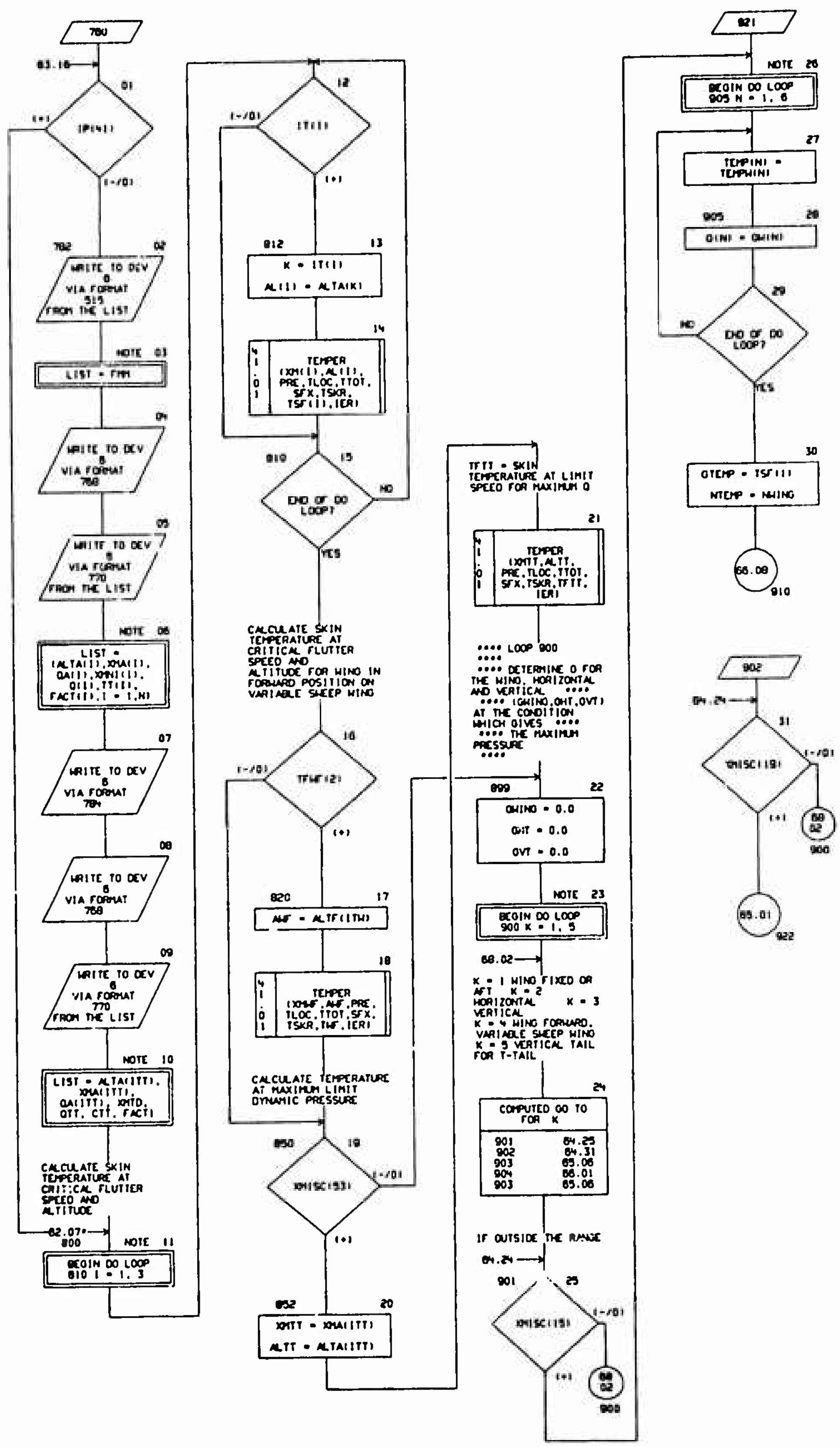


CHART TITLE - SUBROUTINE WHVOO



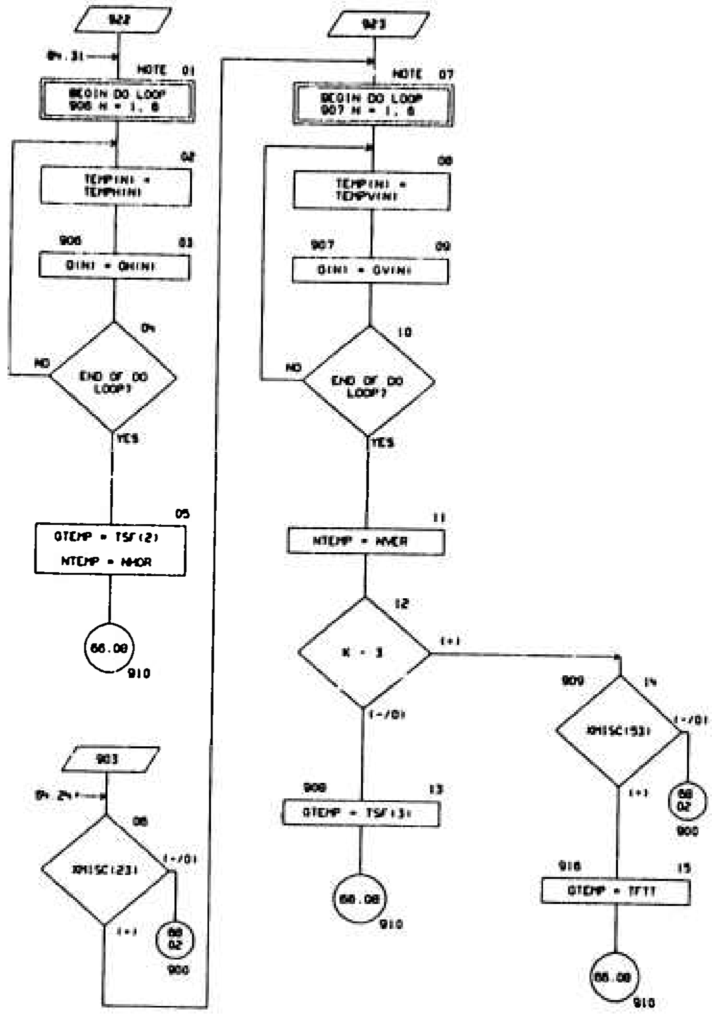
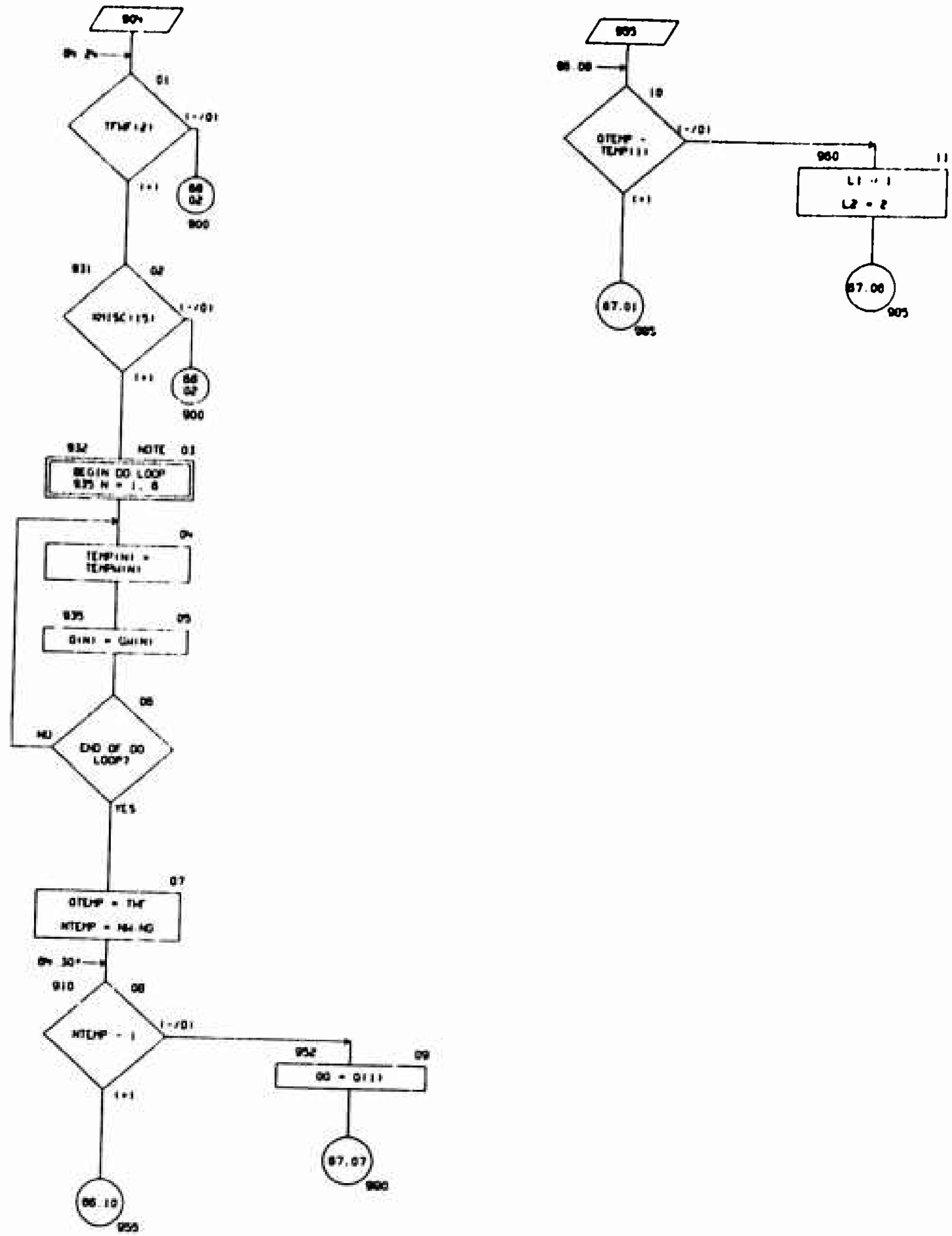

922
NOTE 01
BEGIN DO LOOP 906 N = 1, 6
02
TEMP(N) = TEMPH(N)
906 03
Q(N) = QH(N)
04
NO
END OF DO LOOP?
YES
05
QTEMP = TSF(2)
NTEMP = NHOR
66.08
910
903
06
XMISC(23)
(-/0)
(+)
88 02
900
923
NOTE 07
BEGIN DO LOOP 907 N = 1, 6
08
TEMP(N) = TEMPV(N)
907 09
Q(N) = QV(N)
10
END OF DO LOOP?
11
NTEMP = NVER
12
K - 3
909 14
XMISC(53)
908 13
QTEMP = TSF(3)
916 15
QTEMP = TFTT

CHART TITLE - SUBROUTINE HHVOO



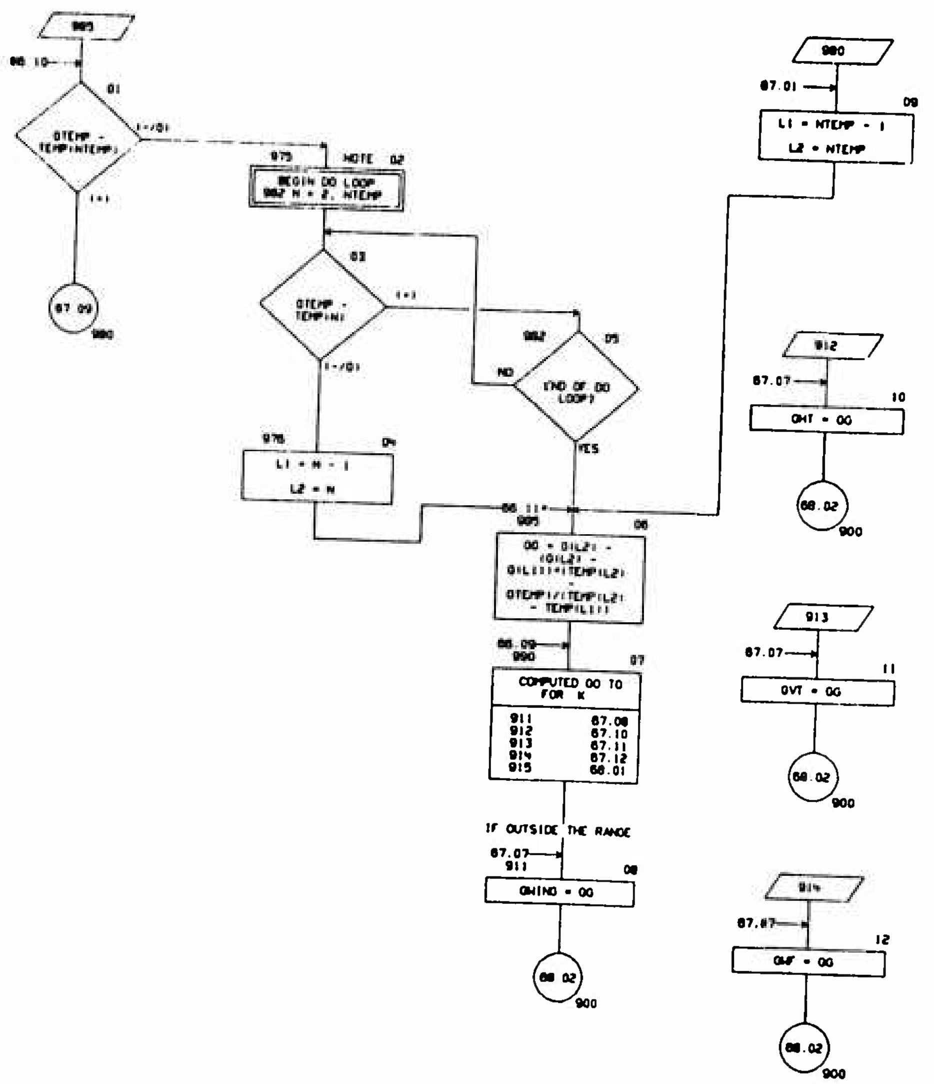


CHART TITLE - SUBROUTINE WHVOO

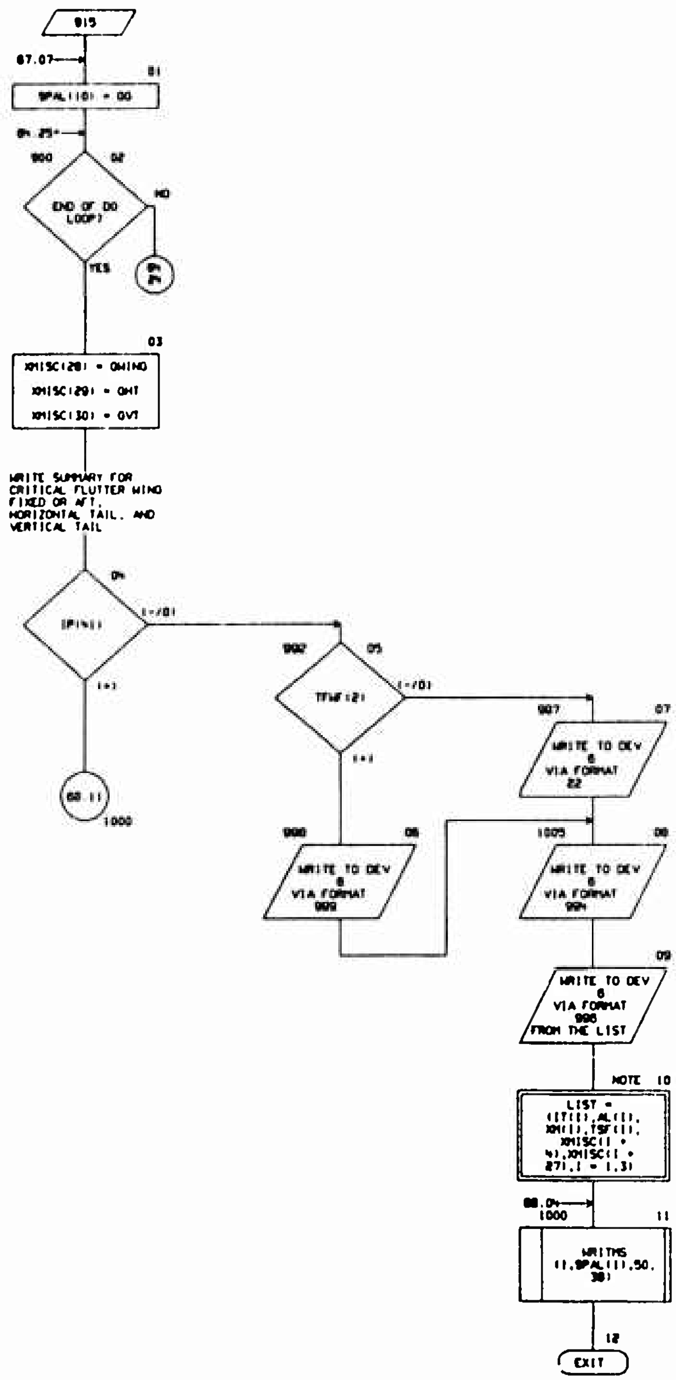


CHART TITLE - NON-PROCEDURAL STATEMENTS

```
          COMMON     SVF(180),DATA(312),TFW(38),TFH(38),TFV(38),TFWF(38),
                     TF(38),XMH(20),Q(20),QQ(20),XMATOP(20),
                     QW(8),QH(8),QV(8)
          COMMON /IPRINT/IP(80)
          COMMON /MISC/XMISC(100)
          DIMENSION TEMPW(8),TEMPH(8),TEMPV(8)
          DIMENSION SPAL(50)
          DIMENSION ALTA(9),XMA(9),ALTF(3),XMF(3)
          DIMENSION GJDAT(100),CTTM(20),CTT1(20),CTT2(20)
          DIMENSION TT(20),FACT(20)
          DIMENSION QA(20)
          DIMENSION AL(3),XM(3),IT(3),TSF(3)
          DIMENSION TEMP(8),Q(8)
          EQUIVALENCE (DATA(305),VFF)
          EQUIVALENCE (SVF(145),TEMPW(1)),(SVF(159),TEMPH(1))
           , (SVF(173),TEMPV(1)),(SVF(151),NWING),(SVF(165),NHOR)
           , (SVF(179),NVER)
          EQUIVALENCE (SPAL(16),FM),(SPAL(17),ALTA(1))
           , (SPAL(26),XMA(1)),(SPAL(35),ALTF(1)),(SPAL(38),XMF(1))
           , (SPAL(41),AMF),(SPAL(42),XMMF),(SPAL(43),TMF)
           , (SPAL(44),QQMF),(SPAL(45),QMF)
          EQUIVALENCE (QTT,SPAL(9)),(CTT,SPAL(11)),(XMTD,SPAL(8))
           ,(TSF(1),SPAL(46)),(TFTT,SPAL(49))
          EQUIVALENCE (PC,GJDAT(18)),(CTTD1,GJDAT(19)),(CTTD2,GJDAT(20))
           , (CTTM(1),GJDAT(21)),(CTT1(1),GJDAT(41)),(CTT2(1),GJDAT(61))
 24       FORMAT(1H1,40X,22HFLUTTER SPEED MARGIN =,F5 2,22X,
            20H** WHVQQ - IP(4)) **)
 412      FORMAT(1H0,45X,17HWING FIXED OR AFT)
 422      FORMAT(1H0,46X,15HHORIZONTAL TAIL)
 432      FORMAT(1H0,47X,13HVERTICAL TAIL)
 442      FORMAT(19X,29HSPEED-ALTITUDE PROFILE POINTS,12X,14HFLUTTER DESIGN/
           82X,12HCOMPRESSIBLE/19X,8HALTITUDE,6X,4HMACH,5X,7HDYNAMIC,11X,
           4HMACH,5X,7HDYNAMIC,9X,7HDYNAMIC / 21X,4HFEET,7X,6HNUMBER,3X,
           8HPRESSURE,10X,6HNUMBER,3X,8HPRESSURE,8X,8HPRESSURE)
 444      FORMAT(16X,F11 0,F11 4,F11 1,5X,F11 4,F11 1,5X,F11.1)
 514      FORMAT(1H1,42X,18HFLUTTER Q MARGIN =,F7.4,22X,
              20H** WHVQQ - IP(4)) **)
 22       FORMAT(1H1,88X,20H** WHVQQ - IP(4)) **)
 606      FORMAT(1H0,48X,12HWING FORWARD)
 25       FORMAT(1H0,40X,22HFLUTTER SPEED MARGIN =,F5 2)
 515      FORMAT(1H0,42X,18HFLUTTER Q MARGIN =,F7 4)
 744        FORMAT( 60H0*****  EXTRAPOLATED ON T-TAIL STIFFNESS COEFF  FOR M
          ACH NO ,   1F6 2,   8H, CTT = , 1E9.3 )
 766      FORMAT(1H0,37X,14HT-TAIL FLUTTER,5X,10HDIHEDRAL =,F5 1)
 768      FORMAT(11X,29HSPEED-ALTITUDE PROFILE POINTS,12X,14HFLUTTER DESIGN/
           11X,8HALTITUDE,6X,4HMACH,5X,7HDYNAMIC,11X,4HMACH,5X,7HDYNAMIC,
            9X,3HCTT,12X,5HQ*CTT/13X,4HFEET,7X,6HNUMBER,3X,8HPRESSURE,
           10X,6HNUMBER,3X,8HPRESSURE)
 770      FORMAT(8X,F11 0,F11.4,F11.1,5X,F11 4,F11 1,2E16.6)
 784      FORMAT(1H0,48X,12HDESIGN POINT/)
 999      FORMAT(//////)
 994      FORMAT(33X,42H*** DESIGN TEMPERATURE, PRESSURE AND Q *** //18X,
           13HPROFILE POINT,4X,8HALTITUDE,3X,8HMACH NO ,3X,11HTEMPERATURE,
           3X,8HPRESSURE,4X,7HQ (PSI) )
 996      FORMAT(8X,4HWING,8X,I7,8X,F11 0,F11 4,F14.1,F11.1,F11 0/
           8X,10HHORIZONTAL,2X,I7,8X,F11 0,F11.4,F14.1,F11.1,F11 0/
           8X,8HVERTICAL,4X,I7,8X,F11.0,F11 4,F14 1,F11.1,F11 0)
```

FORTRAN LISTING

OF

FLUTTER AND TEMPERATURE MODULE

CARD NO **** CONTENTS ****

```
      PROGRAM OLAY3
C
      COMMON    SVF(180),DATA(312),TFH(38),TFH(38),TFV(38),TFWF(38),
     *          TF(38),XMN(20),Q(20),QQ(20),XMATOP(20),
     *          OH(6),OH(6),OV(6)
C
      DIMENSION PS(23),TLOCAL(23),TTOTAL(23),SFLUX(23),TSKINR(23),
     *          TSKINF(23)
C
      DIMENSION BC(195),ALT(23),XMACH(23),BB(10)
C
      EQUIVALENCE        (SVF(  1),PS    (1)),(SVF( 24),TLOCAL(1)),
     * (SVF( 47),TTOTAL(1)),(SVF( 70) SFLUX (1)),(SVF( 93),TSKINR(1)),
     * (SVF(116),TSKINF(1))
C
      DO 10 N=1,180
   10 SVF(N) = 0.0
C
      DO 11 N=1,6
      OH(N) = 0.0
      OH(N) = 0.0
   11 OV(N) = 0.0
C
      CALL READMS(1,BC(1),195,22)
      CALL READMS(1,DATA(1),312,12)
C
C
      DO 123 I=1,23
C
C     **** COMPUTE THE ALTITUDE AND MACH NUMBER AT EACH CONDITION ****
C
      GO TO (72,73,74,75,76,77,76,78,84,77,85,78,86,77,85,76,86,77,85,72
     1,74,72,74),I
C
   72 BB(9)=BC(19)
      BB(10)=BC(166)
      GO TO 87
C
   73 BB(9)=BC(20)
      BB(10)=BC(167)
      GO TO 87
C
   74 BB(9)=BC(21)
      BB(10)=BC(168)
      GO TO 87
C
   75 BB(9)=0.0
      BB(10)= 90
      GO TO 87
C
   76 BB(9)=BC(25)
      BB(10)=BC(28)
      GO TO 87
C
   77 BB(9)=BC(19)
      BB(10)=BC(22)
      GO TO 87
C
   78 BB(10)=1.5*BC(31)/661.3
   81 BB(9)=0.0
      GO TO 87
C
   84 BB(10)=1.2*BC(32)/661.3
      GO TO 81
C
   85 BB(9)=BC(20)
      BB(10)=BC(23)
      GO TO 87
C
   86 BB(9)=BC(26)
```

CARD NO **** CONTENTS ****

```
      BB(10)=BC(29)
C
   87 ALT(I) = BB(9)
      XMACH(I) = BB(10)
C
      IF(XMACH(I))123,123,88
C
C     **** DETERMINE THE PRESSURE, LOCAL AND TOTAL TEMPERATURES,  ****
C     **** SUN FLUX, AND SKIN TEMPERATURE FOR THIS LOAD CONDITION ****
C
   88 CALL TEMPER(XMACH(I),ALT(I),PS(I),TLOCAL(I),TTOTAL(I),SFLUX(I),
     *            TSKINR(I),TSKINF(I),IER)
C
      IF(IER -15)123,123,701
C
  701 WRITE(6,703)I,TSKINF(I)
  703 FORMAT(15H0LOAD CONDITION,I3,77H - THE TEMPERATURE LOOP DID NOT CL
     *OSE IN 100 ITERATIONS. THE SKIN TEMPERATURE,F8 3,27H IS FROM THE L
     *AST ITERATION/)
C
  123 CONTINUE
C
C
C        CALL WHVOO TO GET FLUTTER COMPRESSIBILITY EFFECTS
      CALL WHVOO
C
C
      CALL WRITHS(I,SVF(1),180,31)
C
      END
C
C )))))))))))))))))))))))))))))))))))))))))))))))))));))))))))))))))))))
C                         FUNCTION HBL
C ((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((
C
C     NOTE-  THIS FUNCTION SUBPROGRAM COMPUTES AERODYNAMIC HEATING      HAHN0010
C     COEFFICIENTS BY THE METHOD OF E.R  VAN DRIEST.  IT IS DESCRIBED INHAHN0015
C     REPORT NA-63-1473.  THIS VERSION IS IN FORTRAN IV                 HAHN0020
C                                                                       HAHN0025
C                                                                       HAHN0030
C     PROGRAM TO COMPUTE BOUNDARY LAYER HEAT TRANSFER COEFFICIENTS      HAHN0035
C                                                                       HAHN0040
C     C.J. MAC MILLER                               26 MARCH,1964       HAHN0045
C                                                                       HAHN0050
C     THE EQUATION IS PROGRAMMED AS FOLLOWS...                          HAHN0055
C                                                                       HAHN0060
C       BOUNDARY LAYER HEAT TRANSFER COEFFICIENT  =                     HAHN0065
C                                                                       HAHN0070
C             HBL(XMACH,PO,TO,TSK,DIST,XLAMDA,REY,METHOD,IER)           HAHN0075
C                                                                       HAHN0080
C        WHERE - XMACH = LOCAL MACH NUMBER, DIMENSIONLESS               HAHN0085
C                   PO = LOCAL PRESSURE, PSF                            HAHN0090
C                   TO = LOCAL TEMPERATURE, DEG RANKINE                 HAHN0095
C                  TSK = SKIN TEMPERATURE, DEG RANKINE                  HAHN0100
C                 DIST = CHARACTERISTIC LENGTH, FEET                    HAHN0105
C                        WITH METHOD 1-2, = LENGTH AFT OF LEADING EDGEHAHN0110
C                        WITH METHOD 3-4, = DIAMETER OF LEADING EDGE  HAHN0115
C               XLAMDA = ANGLE OF SWEEP, DEGREES                        HAHN0125
C                  REY = TRANSITION REYNOLDS NUMBER, DIMENSIONLESS, IF HAHN0130
C                        NO VALUE OF REY IS SUBMITTED, THE FUNCTION    HAHN0135
C                        WILL SUPPLY A VALUE OF ONE MILLION             HAHN0140
C               METHOD = SURFACE ORIENTATION                            HAHN0145
C                        METHOD = 1 FOR FLAT PLATE, WEDGES, CYLINDERS  HAHN0150
C                                 ALIGNED WITH FLOW                     HAHN0155
C                               = 2 FOR CONES OR OTHER SURFACES OF REV-HAHN0160
C                                 OLUTION                               HAHN0165
C                               = 3 FOR STAGNATION HEAT TRANSFER OF TWOHAHN0170
C                                 DIMENSIONAL SHAPES AS CYLINDERS      HAHN0175
C                               = 4 FOR STAGNATION HEAT TRANSFER OF    HAHN0180
C                                 THREE DIMENSIONAL SHAPES AS SPHERESHAHN0185
C                  IER = ERROR INDICATOR                                HAHN0190
```

```
CARD NO   ****                      CONTENTS                           ****
          C              IF IER = 0, NO ERROR, SOLUTION CRITERIA WERE MET.  HAHN0195
          C                 IER = 1, SOLUTION FOR TOTAL TEMPERATURE DID NOT HAHN0200
          C                          CONVERGE IN 100 PASSES. VALUE RETURNED HAHN0205
          C                          IS BASED ON CONSTANT GAMMA.            HAHN0210
          C                 IER = 2, SOLUTION FOR BOUNDARY LAYER TEMPERATUREHAHN0215
          C                          DID NOT CONVERGE IN 100 PASSES. VALUE  HAHN0220
          C                          RETURNED IS BASED ON RECOVERY OF  875  HAHN0225
          C              IF IER = 3, SOLUTION FOR SKIN FRICTION COEFFICIENT HAHN0230
          C                          DID NOT CONVERGE IN 100 PASSES. SOLU-  HAHN0235
          C                          TION FOR BOUNDARY LAYER COEF  HAS CON- HAHN0240
          C                          TINUED WITH CURRENT VALUE OF C(F)      HAHN0245
          C                 IER = 4, ARGUMENT XMACH IS ZERO, DIST IS ZERO,  HAHN0250
          C                          HBL = 6 00                             HAHN0255
          C                 IER = 5, ARGUMENT XMACH IS ZERO, DIST IS        HAHN0256
          C                          POSITIVE, HBL = 6 00                   HAHN0257
          C                 IER = 6, ARGUMENT XMACH IS POSITIVE, DIST IS    HAHN0258
          C                          MINUS OR ZERO, HBL = 6.00              HAHN0259
          C                IER = 7, ARGUMENT TO IS BELOW 160 DEGREES RANKINEHAHN0261
          C                A VALUE OF HBL =6 0 IS RETURNED                  HAHN0262
          C                IER = 8, ARGUMENT TSK IS BELOW 160 RANKINE.      HAHN0263
          C                SOLUTION CONTINUES USING TO FOR TSK              HAHN0264
          C                                                                 HAHN0265
          C     THE VALUE OF BOUNDARY LAYER HEAT TRANSFER COEFFICIENT IS RETURNED HAHN0266
          C     IN  BTU/HR-FT**2-DEG F  TO THE CALLING PROGRAM              HAHN0270
          C                                                                 HAHN0275
          C                                                                 HAHN0280
          C     ****                       ***                       **** HAHN0310
                FUNCTION HBL(XMACH,PO,TO,TSK DIST,XLAMDA,REY,METHOD,IER,TOTI,TAWI)HAHN0315
                IF ( XMACH )10,10,13                                        HAHN0320
             10 IF ( DIST )11,11,12                                         HAHN0322
             11 IER = 4                                                     HAHN0324
                GO TO 15                                                    HAHN0325
             12 IER = 5                                                     HAHN0326
                GO TO 15                                                    HAHN0327
             13 IF ( DIST )14,14,20                                         HAHN0328
             14 IER = 6                                                     HAHN0332
             15 HBL = 6 00                                                  HAHN0334
                GO TO 1000                                                  HAHN0335
             20 IF ( TO - 160 0 )21,21,22                                   HAHN0336
             21 IER = 7                                                     HAHN0337
                GO TO 15                                                    HAHN0338
             22 IF ( TSK - 160 0 )23,23,24                                  HAHN0339
             23 IER = 8                                                     HAHN0340
                TSK = TO                                                    HAHN0341
             24 TOT = TOTI                                                  HAHN0343
                METOD = 1                                                   HAHN0345
                IF(METHOD-3)26,25,25                                        HAHN0350
             25 METOD = 2                                                   HAHN0355
             26 TAW = TAWI                                                  HAHN0360
                TSTR =  28*TO +  5*TSK +  22*TAW                            HAHN0365
                GRHO = PO/53 35045/TSTR                                     HAHN0370
                EXP1 =  EXP(5526 /TSTR)                                     HAHN0375
                SPHT = .06856*(3 5 + (5526 /TSTR/(EXP1-1 ))**2 * EXP1)      HAHN0380
                GVIS =  00262*TSTR**1 5/(TSTR+198 7)                        HAHN0385
                COND = .00114*TSTR** 5/(1 +441.2/TSTR/10.**(21.6/TSTR))     HAHN0390
                PRN = SPHT*GVIS/COND                                        HAHN0395
                EXPO =  EXP(5526 /TO)                                       HAHN0400
                SPHTO =  06856*(3.5 + (5526./TO/(EXPO-1.))**2 * EXPO)       HAHN0405
                GAMMAO = SPHTO/(SPHTO-53 35045/778 26)                      HAHN0410
                SPSOO =  SQRT(GAMMAO*32.174*53.35045*TO)                    HAHN0415
                VELO = SPSOO*XMACH                                          HAHN0420
                GRHOO = PO/53.35045/TO                                      HAHN0425
                GVISO = .002627*TO**1.5/(TO+198.7)                          HAHN0430
                GO TO (100,100,300,300),METHOD                              HAHN0435
            100 IF(REY)30,30,35                                             HAHN0440
             30 REY = 1.E6                                                  HAHN0445
             35 RENO = VELO*DIST*GRHOO*3600./GVISO                          HAHN0450
                IF(RENO-REY)40,60,60                                        HAHN0455
             40 REN = VELO*DIST*GRHO*3600./GVIS                             HAHN0460
                ANAL = 1./PRN**.666                                         HAHN0465
                CSFN = .664/ SQRT(REN)                                      HAHN0470
```

```
CARD NO    ****                    CONTENTS                              ****
             HBL = CSFN/2.*ANAL*ORHOO*SPHT*VELO*3600.                    HAHN0475
             IF(METHOD-1)1000,1000,50                                    HAHN0480
          50 HBL = HBL*1.732                                             HAHN0485
             00 TO 1000                                                  HAHN0490
          60 ASQ = (TOT-TO)/TSK                                          HAHN0495
             B = TOT/TSK-1.                                              HAHN0500
             THETA = B/ SORT(B**2+4.*ASQ)                                HAHN0505
             PSI = (2.*ASQ-B)/ SORT(B**2+4.*ASQ)                         HAHN0510
             DENOM = 0 2+2*( ASIN(THETA)+ ASIN(PSI))                     HAHN0515
             CSFNI = .001                                                HAHN0520
             CSFI = 0.                                                   HAHN0525
             IF(METHOD-1)70,70,65                                        HAHN0530
          65 RENO = RENO/2.                                              HAHN0535
          70 ANUM = .41 + ALOG10(RENO*CSFNI*(TO/TSK)**.76)               HAHN0540
             CSFN = .25*CSFNI+.75*TO/ASQ/TSK/(ANUM/DENOM)**2             HAHN0545
             IF( ABS(CSFN-CSFNI)/CSFNI-.001)110,110,80                   HAHN0550
          80 CSFNI = CSFN                                                HAHN0555
             CSFI = CSFI+1.                                              HAHN0560
             IF(CSFI-100.)70,70,90                                       HAHN0565
          90 IER = 3                                                     HAHN0570
         110 HBL = .8*CSFN*ORHOO*SPHT*VELO*3600.                         HAHN0575
             00 TO 1000                                                  HAHN0580
         300 CONDO =  00114*TO**.5/(1.+441.2/TO/10.**(21.6/TO))          HAHN0585
             PRNO = SPHTO*OVISO/CONDO                                    HAHN0590
             XMULT = COS(XLAMDA * .01745323)                             HAHN0595
             IF(XMULT-.5)130,150,150                                     HAHN0600
         130 XMULT = 0.5 - ( ACOS(XMULT)*180.0/3.1416 - 60.0)*0.01       HAHN0605
             IF(XMULT-.35)140,150,150                                    HAHN0610
         140 XMULT = .35                                                 HAHN0615
         150 F2 = .57/PRNO**.6*XMULT                                     HAHN0620
             IF(METHOD-3)170,170,160                                     HAHN0625
         160 F2 = .763/PRNO**.6                                          HAHN0630
         170 EXPO =  EXP(5526./TO)                                       HAHN0635
             EXPTO =  EXP(5526./TOT)                                     HAHN0640
             SPHTI = .06856/(TOT-TO)*(3.5*(TOT-TO)+5526./(EXPTO-1.)-5526./(EXPOHAHN0645
            1-1.))                                                       HAHN0650
             OAMMAI = SPHTI/(SPHTI-53.35045/778.26)                      HAHN0655
             SPSDTT =  SORT(OAMMAI*32.174*53.35045*TOT)                  HAHN0660
             IF(XMACH-1.)180,180,220                                     HAHN0665
         180 VELI = XMACH*SPSDTT                                         HAHN0670
             BETA = 4.*VELI/DIST                                         HAHN0675
             IF(METHOD-3)210,210,190                                     HAHN0680
         190 BETA = .75*BETA                                             HAHN0685
         210 ORHOTT = ORHOO/(1./(1.+(OAMMAI-1.)/2.*XMACH**2))**(1./(OAMMAI-1.))HAHN0690
             00 TO 230                                                   HAHN0695
         220 X2MACH = XMACH*XMACH                                        HAHN0700
             XMACHI = SORT( ( 1.000 +(OAMMAI -1.000 )/2.000*X2MACH )/( OAMMAI   HAHN0705
            1   *X2MACH - ( OAMMAI - 1.000 )/2.00 ))                     HAHN0710
             VELI = XMACHI*SPSDTT                                        HAHN0715
             XAMDA = 1.+2.*((XMACH/XMACHI)**2-1.)/(1.+OAMMAI*X2MACH)     HAHN0720
             BETA = 2.*VELI* SORT(XAMDA)/DIST                            HAHN0725
             RPOPI = ( 1.000 + OAMMAI*XMACHI**2 )*( 1.000 + (OAMMAI-1.000)/2.00HAHN0730
            1   *X2MACH)/( 1.000 + OAMMAI*X2MACH )/( 1.000 + ( OAMMAI - 1.000 )HAHN0735
            2   /2.00*XMACHI**2 )                                        HAHN0736
             RPIPS = (1./(1.+(OAMMAI-1.)/2.*XMACHI**2))**(1./(OAMMAI-1.))   HAHN0740
         229 ORHOTT = ORHOO/RPOPI/RPIPS                                  HAHN0741
         230 CONTINUE                                                    HAHN0750
             OVISTT = 0.00262*TOT**1.5/(TOT+198.7)                       HAHN0751
             EXPSK =  EXP(5526./TSK)                                     HAHN0755
             SPHTSK = .06856*(3.5+(5526./TSK/(EXPSK-1.))**2*EXPSK)       HAHN0760
             HBL = F2* SORT(BETA*3600.*ORHOTT*OVISTT)*SPHTSK             HAHN0765
        1000 RETURN                                                      HAHN0770
             END                                                         HAHN0775
       C
       C )))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))
       C                           FUNCTION PRESH
       C (((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((
       C
             FUNCTION PRESH(ALT)                                         00000040
       C     PRESH IS A SUBROUTINE THAT CONVERTS ALTIMETER READINOS TO PRESSURE00000010
       C     IT IS 0000 FOR ALTIMETERS CALIBRATED IN OEOPOTENTIAL ALTITUDE PER 00000020
```

```
CARD NO  ****                        CONTENTS                           ****
         C     THE RELATION DEFINED BY U.S  STANDARD ATMOSPHERE, 1962     00000030
               ALOFT = ALT/1000.0                                         00000050
               IF ( ALT  - 36089 239)10,30,20                             00000060
            10 PRESH = 2116 22*( ( 1 000 - 6.87559E-3*ALOFT )**5 25591 )  00000070
               GO TO 100                                                  00000080
            20 IF( ALT - 65616 60)30,30,40                                00000090
            30 ZEXPO = EXP( ( ALT - 36089 24 )/20805 56 )                 00000100
               PRESH = 472 68/ZEXPO                                       00000110
               GO TO 100                                                  00000120
            40 IF( ALT - 104986 88 )50,50,60                              00000130
            50 PRESH = 114 345*(( 1.000 + 0 54864*( ALOFT - 65 61680)/389 970 )  00000140
             1    **(-34.16341)                                           00000145
               GO TO 100                                                  00000150
            60 PRESH = 18 131*(1.00+1 536*(ALOFT-104 987)/411.57)**(-12 2012)  00000151
               IF( ALT - 154199.48 )100,100,65                            00000152
            65 WRITE(6,70)                                                00000155
            70 FORMAT(1H 8X49HWARNING- ALTITUDE IS BEYOND VALID RANGE OF PRESH.) 00000160
           100 RETURN                                                     00000170
               END                                                        00000180
         C
         C )))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))
         C                        SUBROUTINE QINC
         C ((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((
         C
               SUBROUTINE QINC (XM,PO,TO,Q)
         C
         C        SUBROUTINE TO CALCULATE INCOMPRESSIBLE DYNAMIC PRESSURE ,Q,PSF
         C        GIVEN, XM = MACH NUMBER              PO = AMBIENT PRESSURE, PSF
         C                TO = AMBIENT TEMPERATURE, DEG R
         C
         C        GIVEN ALTITUDE, PO IS OBTAINED FROM FUNCTION ROUTINE PRESH
         C                    AND TO IS OBTAINED FROM FUNCTION ROUTINE TEMALT
         C        G = ACCELERATION OF GRAVITY, FT/SEC**2
               G = 32.17405
         C        R = GAS CONSTANT, FT/DEG R
               R = 53.35045
         C        5526.0 IS THE CHARACTERISTIC TEMPERATURE OF MOLECULAR
         C        VIBRATION, DEG R
               TOT = 5526.0/TO
               EXPI = EXP(TOT)
         C
         C        SPECIFIC HEAT CALCULATION IS BASED ON EQUATION BY DONALDSON,
         C        COLEMAN DU PONT, NACA DOCUMENT RM NO. L8J14, DECEMBER 10, 1948.
         C        SPHT = CONSTANT PRESSURE SPECIFIC HEAT, FT-LB/LB/DEG R
               SPHT = R * (3.5 + (TOT/(EXPI - 1.0))**2 * EXPI)
               GAMMA = SPHT/(SPHT - R)
         C        SPSD = SPEED OF SOUND, FT/SEC
               SPSD = SQRT(GAMMA * G * R * TO)
         C        RHO = DENSITY OF AIR, SLUGS/FT**3
               RHO = PO/R/TO/G
               Q = RHO/2.0 * (SPSD * XM)**2
               RETURN
               END
         C
         C )))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))
         C                        SUBROUTINE QSUB
         C ((((((((((((((((.(((((((((((((((((((((((((((((((((((((((((((((((((((((((
         C
               SUBROUTINE QSUB(LIM)
         C
               COMMON     SVF(180),DATA(312),TFM(38),TFH(38),TFV(38),TFWF(38),
              *           TF(38),XMNI(20),Q(20),QQ(20),XMATOP(20)
         C
               DIMENSION TABLE(38)
         C
               EQUIVALENCE (TABLE(1),DATA(1))
         C
               DO 100 I=1,LIM
         C
               XMN = XMNI(I)
         C
```

CARD NO **** CONTENTS ****

```
      IF(XMN - TABLE(1))47,47,48
C
   47 TFANS = XMN * TF(1) / TABLE(1)
      XMATOP(I) = XMN
      GO TO 89
C
   48 DO 50 N=2,38
      IF(N - 38)49,52,52
   49 IF(XMN - TABLE(N))52,52,50
   52 NSAVE = N - 1
      TFP = TF(N-1) + (XMN-TABLE(N-1)) / (TABLE(N)-TABLE(N-1))
     *    * (TF(N)-TF(N-1))
      GO TO 54
   50 CONTINUE
C
   54 TOP = 0.0
C
      DO 60 N=1,NSAVE
      IF(TF(N) - TOP)60,60,62
   62 TOP = TF(N)
      NS = N
   60 CONTINUE
C
      IF(TOP - TFP)71,71,72
C
   71 TFANS = TFP
      XMATOP(I) = XMN
      GO TO 89
C
   72 TFANS = TOP
      XMATOP(I) = TABLE(NS)
C
   89 QQ(I) = ((TFANS/XMN) / (TF(4)/TABLE(4)))**2 * Q(I)
C
  100 CONTINUE
C
      RETURN
C
      END
C
C )))))))  ')))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))
C                             FUNCTION SOLARG
C ((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((
C
      FUNCTION SOLARG(ALT)                                              00001000
C     SOLAR IS A SMALL FUNCTION SUBPROGRAM TO GET SOLAR ILLUMINATION (BT00008000
C     U/HOUR/SQUARE FOOT)                                               00009000
C     AS A FUNCTION OF ALTITUDE(FEET)                                   00010000
      ALOFT=ALT/1000.0                                                  00002000
      IF(ALT-30000.0)  10,20,20                                         00003000
   10 SOLARG=435.0-.141*(30.0-ALOFT)**1.85                              00004000
      GO TO 30                                                          00004005
   20 SOLARG=435.0                                                      00005000
   30 RETURN                                                            00006000
      END                                                               00007000
C
C ))))))))))))))))')))))))))))))))))))))))))))))))))))))))))))))))))))))))))
C                          SUBROUTINE SVFTAD
C ((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((
C
      SUBROUTINE SVFTAB
C
C        THIS SUBROUTINE SETS UP TABLES OF FLUTTER PARAMETER VERSUS
C        MACH NUMBER BY INTERPOLATING TABLE DATA FOR SPECIFIC AREA,
C        ASPECT RATIO, AND SWEEP OF C/4.
C        TABLES ARE SET UP FOR WING FIXED OR AFT, WING FORWARD
C        (VARIABLE SWEEP), HORIZONTAL TAIL, AND VERTICAL TAIL
C
      COMMON    SVF(180),DATA(312),TFW(38),TFH(38),TFV(38),TFWF(38)
C
      COMMON /IPRINT/IP(80)
```

```
CARD NO    ****                    CONTENTS                    ****
            COMMON /MISC/ XMISC(100)
      C
            DIMENSION TABLE(30),TBP(30),TAR2(30),TAR6(30),
           * TS80(30),TS860(30),TTR0(30),TTR60(30)
      C       PROGRAM REGION VARIABLES
            DIMENSION TAR(30),TS0(30),TTR(30),TF(30)
      C
      C
      C
      C
            EQUIVALENCE (TABLE(1),DATA(  1)),(  TBP(1),DATA( 39)),
           *           ( TAR2(1),DATA( 77)),( TAR6(1),DATA(115)),
           *           ( TS80(1),DATA(153)),(TS860(1),DATA(191)),
           *           ( TTR0(1),DATA(229)),(TTR60(1),DATA(267))
      C
      C
      C
            DO 699 N=1,39
            TFH(N) = 0.0
            TFH(N) = 0.0
            TFW (N) = 0 0
        899 TFV(N) = 0.0
      C
            ARH   = XMISC(12)
            SBH   = XMISC(13)
            TRH   = XMISC(14)
            ARH   = XMISC(16)
            SBH   = XMISC(17)
            TRH   = XMISC(18)
            ARV   = XMISC(20)
            SBV   = XMISC(21)
            TRV   = XMISC(22)
            ARWF = XMISC(25)
            SBWF = XMISC(26)
            TRWF = XMISC(27)
      C
      C
            DO 300 I=1,4
      C
            GO TO (301,302,303,304),I
      C
        301 IF(ARH)300,300,305
      C
        305 AR = ARH
            SB = SBH
            TR = TRH
            GO TO 308
      C
      C
        302 IF(ARWF)300,300,332
        332 AR = ARWF
            SB = SBWF
            TR = TRWF
            GO TO 308
      C
        303 IF(ARH)300,300,306
      C
        306 AR = ARH
            SB = SBH
            TR = TRH
            GO TO 308
      C
        304 IF(ARV)300,300,307
      C
        307 AR = ARV
            SB = SBV
            TR = TRV
      C
        308 IF(AR - 4.0)200,200,210
        200 IF(AR - 2.0)201,201,204
        201 PCNT = AR / 2.0
```

CARD NO **** CONTENTS ****

```
      DO 202 N=1,38
  202 TAR(N) = PCNT * TAR2(N)
      GO TO 220
  204 PCNT = (AR - 2.0) / 2.0
      DO 205 N=1,38
  205 TAR(N) = TAR2(N) + PCNT * (TBP(N) - TAR2(N))
      GO TO 220
  210 PCNT = (AR - 4.0) / 2.0
      DO 215 N=1,38
  215 TAR(N) = TBP(N) + PCNT * (TAR6(N) - TBP(N))
C
  220 IF(SB - 45.0)230,230,240
  230 PCNT = SB / 45.0
      DO 235 N=1,38
  235 TSB(N) = TSB0(N) + PCNT * (TBP(N) - TSB0(N))
      GO TO 250
  240 PCNT = (SB - 45.0) / 15.0
      DO 245 N=1,38
  245 TSB(N) = TBP(N) + PCNT * (TSB60(N) - TBP(N))
C
  250 IF(TR - .30)260,260,270
  260 PCNT = TR / .30
      DO 265 N=1,38
  265 TTR(N) = TTR0(N) + PCNT * (TBP(N) - TTR0(N))
      GO TO 280
  270 PCNT = (TR - .30) / .30
      DO 275 N=1,38
  275 TTR(N) = TBP(N) + PCNT * (TTR60(N) - TBP(N))
C
  280 DO 290 N=1,38
  290 TF(N) = TAR(N) * TSB(N) * TTR(N) / TBP(N)**2
C
      IF(IP(4))5003,5003,5004
 5003 CONTINUE
C
      GO TO (291,292,293,294),I
C
  291 WRITE(6,295)
  295 FORMAT(1H1,40X,27H*** WING (FIXED OR AFT) ***,
     1 21X,21H** SWFTAB - IP(4)) **//)
      GO TO 298
C
  292 WRITE(6,335)
  335 FORMAT(1H1,40X,22H*** WING (FORWARD) ***,
     1  28X,21H** SWFTAB - IP(4)) **//)
      GO TO 298
C
  293 WRITE(6,296)
  296 FORMAT(1H1,35X,23H*** HORIZONTAL TAIL ***,
     1  30X,21H** SWFTAB - IP(4)) **//)
      GO TO 298
C
  294 WRITE(6,297)
  297 FORMAT(1H1,35X,21H*** VERTICAL TAIL ***,
     1  32X,21H** SWFTAB - IP(4)) **//)
C
  298 WRITE(6,55)AR,SB,TR
   55 FORMAT(1H ,35X,32HFLUTTER PARAMETER VS MACH NUMBER//
     *           18X,8HMACH NO.,12X,5HAR = ,F5.2,4X,13HSWEEP(C/4) = ,
     * F4.1,4H DEG,3X,8HTAPER = ,F4.3,10X,9HCOMPOSITE//)
C
      DO 60 N=1,38
   60 WRITE(6,62)DATA(N),TAR(N),TSB(N),TTR(N),TF(N)
   62 FORMAT(17X,F5.3,18X,F5.4,15X,F5.4,15X,F5.4,15X,F5.4)
C
 5004 CONTINUE
C
      GO TO (321,322,323,324),I
C
  321 DO 325 N=1,38
  325 TFW(N) = TF(N)
```

```
CARD NO   ****                        CONTENTS                         ****
      GO TO 300
C
  322 DO 330 N=1,30
  330 TFWF(N) = TF(N)
      GO TO 300
C
  323 DO 326 N=1,30
  326 TFH(N) = TF(N)
      GO TO 300
C
  324 DO 327 N=1,30
  327 TFV(N) = TF(N)
C
  300 CONTINUE
C
      RETURN
      END
C
C ))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))
C                          FUNCTION TBL
C ((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((
C
C     NOTE-  THIS FUNCTION SUBPROGRAM IS DESCRIBED IN REPORT NA-63-1473.TAWN0010
C     NOTE-  THIS VERSION OF THE FUNCTION IS IN FORTRAN IV               TAWN0015
C                                                                        TAWN0020
C                                                                        TAWN0025
C              PROGRAM TO COMPUTE BOUNDARY LAYER TEMPERATURE             TAWN0030
C                                                                        TAWN0035
C              C.J.MAC MILLER                  12 MARCH,1964             TAWN0040
C                                                                        TAWN0045
C     THE EQUATION IS PROGRAMMED AS FOLLOWS....                          TAWN0050
C                                                                        TAWN0055
C        BOUNDARY LAYER TEMPERATURE =                                    TAWN0060
C                                                                        TAWN0065
C             TBL(XMACH,PO,TO,TSK,DIST,XLAMDA,REY,METHOD,IER,TOTI)       TAWN0070
C                                                                        TAWN0075
C        WHERE - XMACH = LOCAL MACH NUMBER, DIMENSIONLESS                TAWN0080
C                   PO = LOCAL PRESSURE, PSF                             TAWN0085
C                   TO = LOCAL TEMPERATURE, DEG RANKINE                  TAWN0090
C                  TSK = SKIN TEMPERATURE, DEG RANKINE                   TAWN0095
C                 DIST = CHARACTERISTIC LENGTH, DISTANCE AFT OF LEADING TAWN0100
C                        EDGE, FEET                                      TAWN0105
C               XLAMDA = ANGLE OF SWEEP,DEGREES                          TAWN0115
C                  REY = TRANSITION REYNOLDS NUMBER, DIMENSIONLESS       TAWN0120
C               METHOD = INDICATOR TO CONTROL TYPE OF SOLUTION:          TAWN0125
C                IF METHOD = 0 OR 1, FUNCTION RETURNS SKIN TBL           TAWN0130
C                  METHOD = 2, FUNCTION RETURNS STAGNATION TBL           TAWN0135
C                  IER = ERROR INDICATOR                                 TAWN0140
C                   IF IER = 0, NO ERROR, SOLUTION CRITERIA WAS MET      TAWN0145
C                     IER = 2, SOLUTION FOR BOUNDARY LAYER TEMP DID      TAWN0165
C                            NOT CONVERGE IN 100 PASSES, VALUE RE-       TAWN0170
C                            TURNED IS BASED ON RECOV OF .875            TAWN0175
C                 TOTI = TOTAL TEMPERATURE                               TAWN0177
C                                                                        TAWN0180
C     THE VALUE OF BOUNDARY LAYER TEMPERATURE IS RETURNED IN DEGREES     TAWN0185
C     RANKINE TO THE CALLING PROGRAM                                     TAWN0190
C                                                                        TAWN0195
C                                                                        TAWN0205
C                                                                        TAWN0210
C     VARIATION OF SPECIFIC HEAT, VISCOSITY, CONDUCTIVITY, AND DENSITY  TAWN0215
C     ARE TAKEN INTO ACCOUNT                                             TAWN0220
C     REFERENCE - C(P) - NACA TM NO. L8J14, 10 DEC, 1948                 TAWN0225
C                 VISCOSITY - WADC TR55-169                              TAWN0230
C                 CONDUCTIVITY - U.S  STANDARD ATM, 1962                 TAWN0235
C                 DENSITY - PERFECT GAS                                  TAWN0240
C     IF NO VALUE OF REY IS SUPPLIED, THE FUNCTION WILL USE ONE MILLION TAWN0245
C                                                                        TAWN0250
      FUNCTION TBL(XMACH,PO,TO,TSK,DIST,XLAMDA,REY,METHOD,IER,TOTI)
      IF(XMACH-0.120,10,20                                               TAWN0260
   10 TBL = TO                                                           TAWN0265
      GO TO 100                                                          TAWN0270
```

```
CARD NO   ****                      CONTENTS                                   ****

       20 IF(REY-0 )30,30,40                                               TAHN0275
       30 REY = 1 E6                                                       TAHN0280
       40 TOT = TOT1                                                       TAHN0285
          EXP1 =  EXP(5526 /T0)                                            TAHN0290
          SPHT =  06856 *  (3 5 + (5526./T0/(EXP1-1 ))**2* EXP1)           TAHN0295
          SPHTV = SPHT - 53 35045/778.26                                   TAHN0300
          OAMMA = SPHT/SPHTV                                               TAHN0305
          SPSO =  SORT(OAMMA*32 174*53 35045*T0)                           TAHN0310
          VEL = SPSO*XMACH                                                 TAHN0315
          ORHO = P0/53 35045/T0                                            TAHN0320
          OVIS = .002627 * T0**1.5/(T0+198 7)                              TAHN0325
          REN = VEL*DIST*ORHO*3600 /OVIS                                   TAHN0330
          IND = 0                                                          TAHN0335
          TBL1 = T0 + .875*(TOT-T0)                                        TAHN0340
       50 TSTR = .28*T0 +  5*TSK +  22*TBL1                                TAHN0345
          EXP1 =  EXP(5526 /TSTR)                                          TAHN0350
          SPHT =  06856 * (3 5 + (5526 /TSTR/(EXP1-1 ))**2*EXP1)           TAHN0355
          OVIS =  00262*TSTR**1.5/(TSTR+198.7)                             TAHN0360
          COND = .00114*TSTR** 5/(1 +441 2/TSTR/10 **(21 6/TSTR))          TAHN0365
          PRN = SPHT*OVIS/COND                                             TAHN0370
          IF (METHOD-1)58,58,52                                            TAHN0375
       52 IF(XLAMDA)54,54,56                                               TAHN0380
       54 TBL = TOT                                                        TAHN0385
          00 TO 100                                                        TAHN0390
       56 RECOV = 1.000 + ( SORT(PRN) - 1 000 )*(SIN(XLAMDA*.01745323))**2 TAHN0395
          00 TO 70                                                         TAHN0400
       58 RECOV = PRN**.333                                                TAHN0405
          IF(REN-REY)60,70,70                                              TAHN0410
       60 RECOV = PRN**.5                                                  TAHN0415
       70 TBL = T0 + RECOV*(TOT-T0)                                        TAHN0420
          IND = IND+                                                       TAHN0425
          IF( ABS(TBL-TBL1)/TBL- 00001 )100,100,80                         TAHN0430
       80 IF(IND-100)85,90,90                                              TAHN0435
       85 TBL1 = TBL                                                       TAHN0440
          00 TO 50                                                         TAHN0445
       90 IER = 2                                                          TAHN0450
          TBL = T0 +  875*(TOT-T0)                                         TAHN0455
      100 RETURN                                                           TAHN0460
          END                                                              TAHN0465
      C
      C ))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))
      C                           FUNCTION TEMALT
      C ((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((
      C
          FUNCTION TEMALT(ALT,TI)                                          00000200
      C                                                                    00000015
      C   TEMALT IS A FUNCTION SUBPROGRAM FOR COMPUTING AIR TEMPERATURE AS A00000020
      C   FUNCTION OF PRESSURE ALTITUDE FOR FOR VARIOUS MODEL ATMOSPHERES. 00000030
      C   TWO PARAMETERS ARE USED IN THE CALLING STATEMENT -               00000040
      C   (1) THE PRESSURE ALTITUDE AS WOULD BE INDICATED BY AN ALTIMETER  00000050
      C   CALIBRATED IN GEOPOTENTAL ALTITUDE FOR U S. STD. ATMOSPHERE,1962, 00000060
      C   (2) AN INDICATOR OF THE MODEL ATMOSPHERE UNDER CONSIDERATION     00000070
      C        USE 1.0 FOR THE U.S. STANDARD ATMOSPHERE, 1962.             00000080
      C        USE 2.0 FOR THE COLD ATMOSPHERE PER TABLE II OF MIL-STD-210A.00000090
      C        USE 3.0 FOR THE HOT ATMOSPHERE PER TABLE III OF MIL-STD-210A.00000100
      C        USE 4.0 FOR POLAR ATMOSPHERE PER TABLE IV OF MIL-STD-210A.  00000110
      C        USE 5.0 FOR TROPICAL ATMOSPHERE PER TABLE V OF MIL-STD-210A. 00000120
      C        NOTE-  THESE INDICATORS CAN BE EITHER NEGATIVE OR POSITIVE. 00000130
          ALOFT = ALT/1000.0                                               00000210
       10 IT = ABS(TI)                                                     00000220
      C   GET INTO PROPER MODEL ATMOSPHERE.                                00000230
       20 IF ( IT- 2 )100,200,30                                           00000240
       30 IF(IT-4)300,400,40                                               00000250
       40 IF(IT-5)500,500,539                                              00000251
      C                                                                    00000255
      C   EQUATIONS FOR THE U.S. STANDARD ATMOSPHERE, 1962                 00000260
      100 IF( ALT - 36089.24 )110,125,120                                  00000270
      110 TEMALT = 518.670 - 3.56616*ALOFT                                 00000280
          00 TO 800                                                        00000290
      120 IF( ALT - 65616.88 )125,125,140                                  00000300
      125 TEMALT = 389.970                                                 00000310
```

```
CARD NO  ****                      CONTENTS                          ****
              GO TO 830                                            00000320
          140 IF( ALT - 104987. )145,160,160                       00000325
          145 TEMALT = 389 970 + 0 546641*( ALOFT - 65.61688 )     00000330
              GO TO 830                                            00000331
          160 TEMALT = 411.570 + 1 53819 * ( ALOFT - 104 987 )     00000335
              IF( ALT - 154200 )830,830,810                        00000340
C                                                                  00000345
C     EQUATIONS FOR THE EXTREME COLD ATMOSPHERE OF MIL-STD-210A    00000350
          200 IF( ALT - 3311 00 )210,225,220                       00000360
          210 TEMALT = 399 7 + 13 591*ALOFT                        00000370
              GO TO 600                                            00000375
          220 IF( ALT - 10744 0 )225,225,230                       00000380
          225 TEMALT = 444 7                                       00000385
              GO TO 600                                            00000386
          230 IF(ALT-30715.0)235,245,240                           00000390
          235 IF(ALT-22000 0)236,237,237                           00000391
          236 SLOPE=3 2511+ 00761*(ALOFT-10 744)+ 000464*((ALOFT-10 744)**2 0) 00000392
              TEMALT=444 7-SLOPE*(ALOFT-10 744)                    00000393
              GO TO 600                                            00000394
          237 SLOPE= 0233*(ALOFT-22.000)+3.4767                    00000395
              TEMALT=406 5-SLOPE*(ALOFT-22 000)                    00000396
              GO TO 600                                            00000410
          240 IF( ALT - 42377 0 )245,245,250                       00000420
          245 TEMALT = 374.7                                       00000430
              GO TO 600                                            00000440
          250 IF(ALT - 47660 )251,251,252                          00000451
          251 SLOPE=5 1198+ 0325*(ALOFT-42.377)                    00000452
              TEMALT=374.7-SLOPE*(ALOFT-42.377)                    00000453
              GO TO 600                                            00000453
          252 IF( ALT - 50583 )253,265,260                         00000454
          253 TEMALT = 334 7 + 3 70*( 50.583 - ALOFT )             00000455
              GO TO 600                                            00000470
          260 IF( ALT - 61087 0 )265,265,270                       00000480
          265 TEMALT = 334 7                                       00000490
              GO TO 600                                            00000500
          270 IF( ALT - 73055 0 )275,280,280                       00000510
          275 SLOPE=3 2352-.06113*(ALOFT-61.087)+ 00057*((ALOFT-61.087)**2 0) 00000520
              TEMALT=334.7+SLOPE*(ALOFT-61 087)                    00000523
              TEMALT = 334 7 + SLOPE*( ALOFT - 61 087 )            00000525
              GO TO 600                                            00000530
          280 IF(ALT-78000)285,290,290                             00000540
          285 SLOPE=- 0137*(ALOFT-73 055)+ 331                     00000542
              TEMALT=365.7-SLOPE*(ALOFT-73 055)                    00000544
              GO TO 600                                            00000546
          290 SLOPE= 334+ 00255*(ALOFT-78 000)                     00000548
              TEMALT=384.4-SLOPE*(ALOFT-78.000)                    00000549
              GO TO 600                                            00000550
C                                                                  00000555
C     EQUATIONS FOR THE EXTREME HOT ATMOSPHERE OF MIL-STD-210A     00000560
          300 IF( ALT - 34000 0 )310,325,320                       00000570
          310 SLOPE=3 942- 00344*ALOFT                             00000580
              TEMALT=562.7-SLOPE*ALOFT                             00000585
              GO TO 600                                            00000590
          320 IF( ALT - 39400.0 )325,335,330                       00000600
          325 SLOPE=.0206*(ALOFT-34 000)+3.2794                    00000610
              TEMALT=432 9-SLOPE*(ALOFT-34.000)                    00000615
              GO TO 600                                            00000620
          330 IF( ALT - 50400 )335,345,340                         00000630
          335 TEMALT = 414.7 + 0 4444*( ALOFT - 39.400 )           00000640
              GO TO 600                                            00000650
          340 IF( ALT - 68400.0 )345,350,350                       00000660
          345 TEMALT = 419.7 + 0.1875*( ALOFT - 50.400 )           00000670
              GO TO 600                                            00000675
          350 IF(ALT-70000)360,370,370                             00000680
          360 SLOPE=.0233*(ALOFT-68.400)+ 5574                     00000682
              TEMALT=422.7+SLOPE*(ALOFT-68.400)                    00000684
              GO TO 600                                            00000686
          370 SLOPE=.6835+.00266*(ALOFT-70.000)+.000001605*((ALOFT-70.000)**2.0)00000687
              TEMALT=425.0+SLOPE*(ALOFT-70.000)                    00000689
              GO TO 600                                            00000690
C                                                                  00000695
```

```
****                              CONTENTS                              ****
C     EQUATIONS FOR POLAR ATMOSPHERE OF MIL-STD-210A.                   00000700
  400 IF( ALT - 3243 )410,425,420                                       00000710
  410 TEMALT=444 0+3 060*ALOFT                                          00000720
      GO TO 600                                                         00000730
  420 IF( ALT - 9802 )425,435,430                                       00000740
  425 TEMALT = 453.9 - 0 543*(ALOFT - 3 243 )                           00000750
      GO TO 600                                                         00000760
  430 IF( ALT - 30065 0 )435,445,440                                    00000770
  435 SLOPE= 000554*(ALOFT-9 802)+2 839                                 00000780
      TEMALT=450 3-SLOPE*(ALOFT-9 802)                                  00000781
      GO TO 600                                                         00000790
  440 IF( ALT - 86092 0 )445,450,450                                    00000800
  445 TEMALT = 392 7 - 0 25703*( ALOFT - 30 065 )                       00000810
      GO TO 600                                                         00000820
  450 TEMALT = 378 3                                                    00000830
      GO TO 600                                                         00000840
C                                                                       00000845
C     EQUATIONS OF THE TROPICAL ATMOSPHERE OF MIL-STD-210A              00000850
  500 IF(ALT-53595 0)501,525,520                                        00000860
  501 IF(ALT-27500)502,503,503                                          00000861
  502 TEMALT=549 5-3 88*(ALOFT)                                         00000862
      GO TO 600                                                         00000863
  503 SLOPE=3 80  00480*(ALOFT-27 500)                                  00000864
      TEMALT=443 0-SLOPE*(ALOFT-27 500)                                 00000865
      GO TO 600
  520 IF( ALT - 69620 0 )525,530,530                                    00000890
  525 IF(ALT-58000)526,527,527                                          00000900
  526 SLOPE=2 222- 029*(ALOFT-53 595)+ 005*((ALOFT-53 595)**2.0)        00000901
      TEMALT=347.7+SLOPE*(ALOFT-53 595)                                 00000902
      GO TO 600                                                         00000903
  527 SLOPE= 00311*(ALOFT-58 000)+2 244                                 00000904
      TEMALT=357 2+SLOPE*(ALOFT-58 000)                                 00000905
      GO TO 600                                                         00000910
  530 SLOPE=. 3087+ 0165*(ALOFT-69 620)-.0000277*((ALOFT-69 620)**2.0)  00000920
      TEMALT=381 7+SLOPE*(ALOFT-69 620)                                 00000923
      GO TO 600                                                         00000924
C                                                                       00000925
C     EQUATIONS FOR DEFROST DAY OF MIL-T-5842                           00000925
  539 ALOFT=ALT/1000 0                                                  00000926
      IF(ALT-50000 0)540,550,550                                        00000927
  540 IF(ALT-25000.0) 542,542,543                                       00000928
  542 TEMALT=543 0-5 97*ALOFT                                           00000929
      GO TO 600                                                         00000930
  543 TEMALT=394 0-1 0*(ALOFT-25.000)                                   00000931
      GO TO 600                                                         00000932
  550 TEMALT=369 0                                                      00000933
C     TEST IF WE ARE WITHIN THE 100000 FEET USED IN DEFINITIONS         00000930
  600 IF(ALOFT - 100 0 )630,630,610                                     00000940
  610 WRITE(6,620)                                                      00000950
  620 FORMAT(1H 6X50HWARNING- ALTITUDE IS BEYOND VALID RANGE OF TEMALT.)00000960
  630 RETURN                                                            00000970
      END                                                               00000980
C
C ))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))
C                             SUBROUTINE TEMPER
C ((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((
C
      SUBROUTINE TEMPER(XMACH,ALT,PRESS1,TLOC,TTOT,SUN,XSKIN,XSKINF,IER)
C
C     SUBROUTINE TEMPER DETERMINES THE PRESSURE, LOCAL TEMPERATURE,
C     TOTAL TEMPERATURE, SOLAR FLUX AND SKIN TEMPERATURE FOR THE SPEED
C     AND ALTITUDE GIVEN IN XMACH AND ALT
C
      COMMON    SVF(180),DATA(312),TFW(38),TFH(38),TFV(38),TFWF(38)
C
      EQUIVALENCE        (EMISS ,DATA(306)),(ABSORP,DATA(307)),
     * (COSPHI,DATA(308)),(ATMOS ,DATA(309)),(XLNOTH,DATA(310)),
     * (TREY  ,DATA(311)),(SWEEP ,DATA(312))
C
C     SET THE INDICATORS TO COMPUTE THE SKIN (NOT THE STAGNATION POINT)
C     TEMPERATURE AND HEAT TRANSFER COEFFICIENT IN TBL AND HBL
```

```
C
      METHOD = 1
      METHO = 1
C
C     INITIALIZE THE ERROR AND PASS INDICATORS TO ZERO
C
      IER = 0
      IPASS = 0
C
C     SET THE TOLERANCE TO ONE-TENTH DEGREE RANKINE
C
      CRIT = .1
C
C     CALL FUNCTION PRESH TO DETERMINE THE PRESSURE AT ALTITUDE
C
      PRESS = PRESH(ALT)
      PRESSI = PRESS/144.0
C
C     CALL FUNCTION TEMALT TO DETERMINE THE LOCAL TEMPERATURE
C     AT ALTITUDE  (ATMOS INDICATES THE  THE ATMOSPHERIC TABLE
C     TO BE USED. A VALUE OF 1.0, WHICH INDICATES STANDARD
C     ATMOSPHERE, IS IN THE PERMANENT DATA IN DATA(309)
C
      TLOC = TEMALT(ALT,ATMOS)
C
C     CALL FUNCTION TTO TO DETERMINE THE TOTAL TEMPERATURE
C
      TTOT = TTO(XMACH,TLOC,IER)
C
C     INITIALIZE THE SKIN TEMPERATURE AT TOTAL TEMP
      XSKIN = TTOT
C
C     CALL FUNCTION SOLARG TO DETERMINE THE SOLAR FLUX  (COSPHI IS
C     THE COSINE OF THE ANGLE TO THE SUN. A VALUE OF 1.0 IS IN
C     THE PERMANENT DATA IN DATA(308)
C
      SUN = SOLARG(ALT) * COSPHI
C
C     START THE ITERATION TO SOLVE THE EQUATION TEMP=A-B*TEMP**4
C     IF THE LOOP DOES NOT CLOSE TO ONE-TENTH OF A DEGREE
C     IN 90 ITERATIONS, THE TOLERANCE IS INCREASED TO ONE DEGREE
C     IF THE LOOP DOES NOT CLOSE IN 10 MORE ITERATIONS, THE ERROR
C     INDICATOR IS SET TO 15 AND THE TEMPERATURE FROM THE LAST
C     ITERATION IS USED
C
C
   59 IPASS = IPASS + 1
C
C     CALL FUNCTION TBL TO DETERMINE THE BOUNDARY LAYER TEMPERATURE
C
      BLT = TBL(XMACH,PRESS,TLOC,XSKIN,XLNGTH,SWEEP,TREY,METHO,IER,TTOT)
C
C     CALL FUNCTION HBL TO DETERMINE THE BOUNDARY LAYER
C     HEAT TRANSFER COEFFICIENT
C
      BLHTC = HBL(XMACH,PRESS,TLOC,XSKIN,XLNGTH,SWEEP,TREY,METHOD,IER,
     *  TTOT,BLT)
C
C     COMPUTE THE VALUES OF A AND B FOR THE EQUATION TEMP=A-B*TEMP**4
C     THE ABSORPTIVITY (ABSORP) AND EMMISSIVITY (EMISS) ARE SET
C     TO .85 IN THE PERMANENT DATA IN DATA(306) AND DATA(307)
C
C     *** NOTE THAT THE TERMS FOR THE HEAT TRANSFERRED TO THE SKIN ***
C     *** FROM A HEAT SINK AND FROM AN EXTERNAL SOURCE, AND THE    ***
C     *** TERMS FOR THE HEAT RADIATED TO A HEAT SINK, HAVE BEEN    ***
C     *** OMITTED FROM THE EQUATIONS FOR A AND B. THIS IS THE SAME ***
C     *** AS INPUTTING A VALUE OF 0.0 IN DD(9),DD(10),DD(11) AND   ***
C     *** DD(15) IN THE ORIGINAL TEMPERATURE PROGRAM OBTAINED      ***
C     *** FROM H. HAROLDSON                                         ***
C
      A = (BLHTC*BLT + SUN*ABSORP) / BLHTC
```

CARD NO **** CONTENTS ****

```
      B = 1714 E-12 * EMISS / BLHTC
C
C     CALL FUNCTION TSKIN TO SOLVE THE EQUATION TEMP=A-B*TEMP**4
C
      XSKIN2 = TSKIN(A,B,CRIT,IER)
C
C     DETERMINE THE DIFFERENCE BETWEEN THE ASSUMED AND
C     COMPUTED TEMPERATURE
C
      XY = ABS(XSKIN2 - XSKIN)
C
C     SET THE ASSUMED TEMPERATURE FOR THE NEXT ITERATION EQUAL TO
C     THE TEMPERATURE COMPUTED IN THIS ITERATION
C
      XSKIN = XSKIN2
C
      IPTEN = IPASS/10
C
      IF(IPTEN - 9)103,101,102
C
  101 CRIT = 1.0
      GO TO 103
C
  102 IER = 15
      GO TO 105
C
C     COMPARE THE DIFFERENCE BETWEEN THE ASSUMED TEMPERATURE AND THE
C     CALCULATED TEMPERATURE WITH THE TOLERANCE
C
  103 IF(XY - CRIT)105,105,59
C
C     DETERMINE THE FINAL TEMPERATURE IN DEGREES F
C
  105 XSKINF = AMAX1(XSKIN - 459.67 , 0.0)
C
      RETURN
      END
C
C ))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))
C                            FUNCTION TSKIN
C ((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((
C
C         TSKIN IS A FUNCTION SUBPROGRAM FOR SOLVING TEMPERATURES FROM
C     AN EQUATION OF FIRST AND FOURTH POWERS IN THE FORM
C         T = A-B* T**4
C     ALL TEMPERATURES ARE IN DEGREES RANKINE  THE ROOT IS REAL AND
C     POSITIVE.
C          THE FUNCTION IS CALLED BY THE STATEMENT
C          TEMP = TSKIN(A,B,C,IER)
C     WHEN A AND B ARE CONSTANTS, C IS A CRITERIAN FOR SATISFACTORY
C     CONVERGENCE, AND IER IS AN ERROR INDICATOR.
C
      FUNCTION TSKIN(A,B,C,IER)
C     TEST OFR POSITIVE VALUES OF A AND B. RETURN IF EITHER IS NEGATIVE.
        IF (A) 2,2,7
    2 IF (B) 3,5,5
    3 IER = 13
      WRITE (6,4)
    4 FORMAT(1H0 10X,'BOTH A AND B ARE NEGATIVE. TSKIN IS DISCONTINUED')
      GO TO 70
    5 IER = 11
      WRITE(6,6)
    6 FORMAT(1H0 10X,'A IS NEGATIVE OR ZERO. TSKIN IS DISCONTINUED.')
      GO TO 70
    7 IF(B) 8,10,15
    8 IER = 12
      WRITE (6,9)
    9 FORMAT(1H0 10X,'B IS NEGATIVE. TSKIN IS DISCONTINUED')
      GO TO 70
   10 TSKIN = A
      GO TO 70
```

CARD NO **** CONTENTS ****

```
   15 XP =(A/B)**0 25                                                    00000380
      YP = 0 0                                                           00000390
      XR =(XP * A)/ (XP + A)                                             00000400
      YR = A * (1.- (A/(XP +A))**4)                                      00000410
C     ICOUNT IS A COUNTER FOR THE NUMBER OF PASSES USED IN CONVERGING.   00000934
      ICOUNT = 0                                                         00000935
C     AT THIS POINT WE START ITERATING                                   00000936
   21 ICOUNT = ICOUNT + 1                                                00000937
      XNUM = XR*YP - XP*YR                                               00000940
      DENOM = XR - XP - YR + YP                                          00000941
   25 IF ( ABS(DENOM)-C)30,30,40                                         00000942
   30 TSKIN = YR                                                         00000943
      GO TO 70                                                           00000944
   40 XS = XNUM / DENOM                                                  00000950
      YS = A - B * XS**4                                                 00000951
C     COMPARE TRIAL (XS) OF SKIN TEMPERATURE WITH COMPUTED VALUE (YS).   00000960
C     IF SATISFACTORILY CLOSE RETURN TOMAIN PROGRAM. OTHERWISE SELLECT   00000961
C     A NEW TRIAL VALUE BASED ON THE TWO PREVIOUS SETS OF NUMBERS.       00000962
   45 IF( ABS(XS-YS) -C)60,60,50                                         00000963
   50 XP = XR                                                            00000970
      YP = YR                                                            00000971
      XR = XS                                                            00000972
      YR = YS                                                            00000973
      IF( ICOUNT - 50 )21,21,55                                          00000974
   55 IER = 16                                                           00000975
   60 TSKIN = YS                                                         00000980
   70 RETURN                                                             00000981
      END
C
C ))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))
C                            FUNCTION TTO
C ((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((
C
C     TOT4,  MACMILLER,  DEPT 381,  GROUP 132,  STATION 2-17.           TOT40005
C     NOTE-  THIS FUNCTION SUBPROGRAM IS DESCRIBED IN REPORT NA-63-1473.TOT40010
C     NOTE-  THIS VERSION OF THE FUNCTION IS IN FORTRAN IV.             TOT40015
C                                                                       TOT4002V
C                                                                       TOT40025
C                       PROGRAM TO COMPUTE TOTAL TEMPERATURE            TOT40030
C                        C.J MAC MILLER    10 MARCH 1964                TOT40035
C                                                                       TOT40040
C     THE EQUATION IS PROGRAMMED AS                                     TOT40045
C                                                                       TOT40050
C         TOTAL TEMPERATURE = TTO(XMACH,TO,IER)                         TOT40055
C                                                                       TOT40060
C          WHERE - XMACH = FREESTREAM MACH NUMBER - DIMENSIONLESS       TOT40065
C                     TO = FREESTREAM STATIC TEMPERATURE - DEG RANKINE  TOT40070
C                   IER = ERROR INDICATOR                               TOT40075
C                     IF IER = 0, NO ERROR, SOLUTION CRITERIA WAS MET   TOT40080
C                       IER = 1,CONVERGENCE HAS NOT ACHIEVED WITHIN 100TOT40085
C                     PASSES, VALUE OF TTO RETURNED IS BASED ON CONSTANTTOT40090
C                     GAMMA                                             TOT40095
C                                                                       TOT40100
C     THE BASIC DERIVATION RELATES CHANGE IN ENTHALPY OF AIR TO CHANGE  TOT40105
C     IN KINETIC ENERGY                                                 TOT40110
C                                                                       TOT40115
C     VARIATION OF SPECIFIC HEAT WITH TEMPERATURE IS TAKEN INTO ACCOUNT TOT40120
C     REFERENCE - NACA TM NO. L8J14, 10DEC.1948                         TOT40125
C                                                                       TOT40130
      FUNCTION TTO(XMACH,TO,IER)                                        TOT40135
      IF(XMACH-0.15)3,3,5                                               TOT40140
    3 TTO = TO                                                          TOT40145
      GO TO 50                                                          TOT40150
    5 EXP1 =  EXP(5526./TO)                                             TOT40155
      SPHT = .06858 *  (3.5 + (5526./TO/(EXP1-1.))**2 * EXP1)           TOT40160
      SPHTV = SPHT - 53.35045/778.26                                    TOT40165
      GAMMA = SPHT/SPHTV                                                TOT40170
      SPSD =  SQRT(GAMMA*32.174*53.35045*TO)                            TOT40175
      VEL = SPSD*XMACH                                                  TOT40180
      SPHT1 = .24                                                       TOT40185
      IND = 0                                                           TOT40190
```

CARD NO **** CONTENTS ****

```
      IER = 0                                                          TOT40195
   10 DELTAT = VEL**2/2./32 174/778 26/SPHTI                           TOT40200
      TT0 = T0 + DELTAT                                                TOT40205
      EXP2 =  EXP(5526./TT0)                                           TOT40210
      SPHTA = 0.06856/DELTAT * ( 3.5*DELTAT + 5526 /( EXP2 - 1.000)    TOT40215
     1   -5526./(EXP1 - 1.000 ))                                       TOT40220
      IND = IND+1                                                      TOT40225
      IF( ABS(SPHTA-SPHTI)/SPHTI-.0001)50,20,20                        TOT40230
   20 IF(IND-100)30,30,40                                              TOT40235
   30 SPHTI = SPHTA                                                    TOT40240
      GO TO 10                                                         TOT40245
   40 TT0 = T0 * (1. + XMACH**2/5.)                                    TOT40250
      IER = 1                                                          TOT40255
   50 RETURN                                                           TOT40260
      END                                                              TOT40265
C
C )))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))
C                          SUBROUTINE WHVMAT
C ((((('((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((
C
      SUBROUTINE WHVMAT
C        THIS SUBROUTINE SETS UP TABLES OF STRESS AND Q VERSUS
C        TEMPERATURE FOR WING, HORRIZONTAL TAIL  AND VERTICAL TAIL
C        MATERIALS
      COMMON     SVF(180),DATA(312),TFW(38),TFH(38),TFV(38),TFWF(38),
     *           TF(38),XMN(20),Q(20),QQ(20),XMATOP(20),
     *           QW(6),QH(6),QV(6)
      COMMON /IPRINT/IP(80)
      COMMON /MISC/XMISC(100)
      DIMENSION  STW(6),STH(6),STV(6),TEMPW(6),TEMPH(6),TEMPV(6)
C        PROGRAM REGION VARIABLES
      DIMENSION TM(300),ST(6),TEMP(6),Q(6)
C
      EQUIVALENCE                                (SVF(139),STW   (1)),
     * (SVF(145),TEMPW (1)),(SVF(153),STH   (1)),(SVF(159),TEMPH (1)),
     * (SVF(167),STV   (1)),(SVF(173),TEMPV (1)),(SVF(151),NWING    ),
     * (SVF(165),NHOR     ),(SVF(179),NVER     ),(SVF(152),S80W     ),
     * (SVF(166),S80H     ),(SVF(180),S80V     )
      DMNOW = XMISC(15)
      DMNOH = XMISC(19)
      DMNOV = XMISC(23)
C
      IND = 0
C
      DO 600 I=1,3
C
      GO TO (601,602,603),I
C
  601 IF(DMNOW)600,600,605
C
  605 IF1 = 40.0 + DMNOW
      GO TO 610
C
  602 IF(DMNOH)600,600,615
C
  615 IF(DMNOH - DMNOW)616,852,616
  616 IF1 = 40.0 + DMNOH
      GO TO 610
C
  603 IF(DMNOV)600,600,625
C
  625 IF(DMNOV - DMNOH)626,852,626
  626 IF1 = 40.0 + DMNOV
C
  610 DO 611 N=1,300
  611 TM(N) = 0.0
C
      CALL READMS(1,TM(1),300,IF1)
C
      DO 840 N=1,6
      NN =  93 + 25*N
```

CARD NO **** CONTENTS ****

```
      ST(N) = TM(NN)
      NN = 175 + 25*N
      TEMP(N) = TM(NN)
      IF(TM(NN+2))638,638,639
  638 Q(N) = 0 0
      GO TO 640
  639 Q(N) = TM(NN+4) / TM(NN+2) / 2 0 / (1 0+TM(NN+1))
  640 CONTINUE
C
C     **** DETERMINE S80, THE STRESS LEVEL AT 80 DEGREES,****
C     **** AND NTEMP, THE NUMBER OF VALUES IN THE        ****
C     **** STRESS AND TEMPERATURE TABLES                 ****
C
      IF(TEMP(2))801,801,805
C
  801 NTEMP = 1
      S80 = ST(1)
      GO TO 850
C
  805 NTEMP = 6
C
      DO 810 N=3,6
      IF(TEMP(N))807,807,810
  807 NTEMP = N - 1
      GO TO 815
  810 CONTINUE
C
  815 IF(80 0 - TEMP(1))820,820,825
C
  820 L1 = 1
      L2 = 2
      GO TO 845
C
  825 IF(80 0 - TEMP(NTEMP))835,835,830
C
  830 L1 = NTEMP - 1
      L2 = NTEMP
      GO TO 845
C
  835 DO 840 N=2,NTEMP
      IF(80 0 - TEMP(N))836,836,840
  836 L1 = N - 1
      L2 = N
      GO TO 845
  840 CONTINUE
C
  845 S80 = ST(L2) - (ST(L2) - ST(L1)) *
     *    (TEMP(L2) - 80 0) / (TEMP(L2) - TEMP(L1))
C
  850 CONTINUE
C
  852 IF(IP(4))5001,5001,5002
 5001 CONTINUE
      IF(IND)669,669,668
  669 WRITE(6,670)
  670 FORMAT(1H1,88X,21H** WHVMAT - IP(4)) **)
      IND = 1
  668 CONTINUE
C
      GO TO(671,672,673),I
C
  671 WRITE(6,685)
  685 FORMAT(   45X,12H*** WING *** )
      GO TO 650
C
  672 WRITE(6,686)
  686 FORMAT(  /40X,23H*** HORIZONTAL TAIL *** )
      GO TO 650
C
  673 WRITE(6,687)
  687 FORMAT(  /40X,21H*** VERTICAL TAIL *** )
```

```
      C
      C
        850 WRITE(6,49)(TM(N),N=285,300)
         49 FORMAT(10X,8A10/10X,8A10//30X,11HTEMPERATURE,5X,12HSTRESS (PSI),
           * 10X,7HQ (PSI)/)
      C
            DO 50 N=1,6
         50 WRITE(6,51)TEMP(N),ST(N),Q(N)
         51 FORMAT(33X,F6.0,9X,F8.0,9X,F11.0)
      C
            WRITE(6,52)S80
         52 FORMAT(/30X,20HSTRESS AT 80 DEGREES,F10.0)
      C
       5002 CONTINUE
      C
            GO TO (681,682,683),I
      C
        681 DO 691 N=1,6
            QW(N) = Q(N)
            STW(N) = ST(N)
        691 TEMPW(N) = TEMP(N)
            NWING = NTEMP
            S80W = S80
            GO TO 600
      C
        682 DO 692 N=1,6
            QH(N) = Q(N)
            STH(N) = ST(N)
        692 TEMPH(N) = TEMP(N)
            NHOR  = NTEMP
            S80H = S80
            GO TO 600
      C
        683 DO 693 N=1,6
            QV(N) = Q(N)
            STV(N) = ST(N)
        693 TEMPV(N) = TEMP(N)
            NVER  = NTEMP
            S80V = S80
      C
        600 CONTINUE
            RETURN
            END
      C
      C ))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))
      C                          SUBROUTINE WHVQQ
      C ((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((
      C
            SUBROUTINE WHVQQ
      C
      C        SUBROUTINE TO CALCULATE COMPRESSIBILITY EFFECTS
      C        FOR WING, HORIZONTAL TAIL, AND VERTICAL TAIL FLUTTER
      C
            COMMON     SVF(180),DATA(312),TFW(38),TFH(38),TFV(38),TFWF(38),
           *           TF(38),XMN1(20),Q(20),QQ(20),XMATOP(20),
           *           QW(6),QH(6),QV(6)
      C
            COMMON /IPRINT/IP(80)
            COMMON /MISC/XMISC(100)
      C
            DIMENSION TEMPW(6),TEMPH(6),TEMPV(6)
      C
            DIMENSION SPAL(50)
            DIMENSION ALTA(9),XMA(9),ALTF(3),XMF(3)
            DIMENSION QJDAT(100),CTTM(20),CTT1(20),CTT2(20)
            DIMENSION TT(20),FACT(20)
      C
            DIMENSION QA(20)
            DIMENSION AL(3),XM(3),IT(3),TSF(3)
            DIMENSION TEMP(6),Q(6)
      C
```

CARD NO **** CONTENTS ****

```
      EQUIVALENCE (DATA(305),VFF)
C
      EQUIVALENCE (SVF(145),TEMPV(1)),(SVF(159),TEMPH(1))
     * , (SVF(173),TEMPV(1)),(SVF(151),NWING),(SVF(165),NHOR)
     * , (SVF(179),NVER)
C
      EQUIVALENCE (SPAL(16),FM),(SPAL(17),ALTA(1))
     * , (SPAL(26),XMA(1)),(SPAL(35),ALTF(1)),(SPAL(38),XMF(1))
     * , (SPAL(41),AHF),(SPAL(42),XMHF),(SPAL(43),THF)
     * , (SPAL(44),OOHF),(SPAL(45),OHF)
      EQUIVALENCE (QTT,SPAL(9)),(CTT,SPAL(11)),(XMTD,SPAL(8))
     * ,(TSF(1),SPAL(46)),(TFTT,SPAL(49))
      EQUIVALENCE (PC,OJDAT(18)),(CTTD1,OJDAT(19)),(CTTD2,OJDAT(20))
     * , (CTTM(1),OJDAT(21)),(CTT1(1),OJDAT(41)),(CTT2(1),OJDAT(61))
C
C        CALL WHVMAT TO GET MATERIAL PROPERTIES FOR CRITICAL LOAD
C        CALCULATION IN OVERLAY 4
C        ALSO FOR CALCULATION OF SHEAR MODULUS AT CRITICAL
C        FLUTTER CONDITIONS
      CALL WHVMAT
C
C        READ SPEED PROFILE RECORD 38
C        RECORD 38 WRITTEN IN DATA MANAGEMENT MODULE
C
      CALL READMS(1,SPAL(1),50,38)
C
C        CALCULATE FLUTTER PARAMETER, CALL SVFTAB
      CALL SVFTAB
      DO 10 I=1,3
      IT(I) = 0
   10 CONTINUE
C
C        TEST IF FLUTTER MARGIN SETUP IN RECORD 38
C        IF NOT USE LIBRARY VALUE
      IF(FM) 12,12,14
   12 FM = VFF
   14 FMM = FM**2
C
      IF(IP(4)) 20,20,30
   20 WRITE(6,24) FM
   24 FORMAT(1H1,40X,22HFLUTTER SPEED MARGIN =,F5 2,22X,
     * 20H** WHVOO - IP(4) **)
C
C        J = 1 PASS THROUGH LOOP 500 FOR MARGIN ON FLUTTER SPEED
   30 J = 1
C
C        N = 9 IS NUMBER OF SPEED ALTITUDE PROFILE POINTS
      N = 7
C
C        CALCULATE INCOMPRESSIBLE Q FOR H(L) DIAGRAM WINGS FIXED OR AFT
      DO 35 I=1,N
      PO = PRESH(ALTA(I))
      TO = TEMALT(ALTA(I),1.0)
      CALL QINC(XMA(I),PO,TO,QA(I))
      XMN1(I) = FM*XMA(I)
      Q(I) = FMM*QA(I)
   35 CONTINUE
C
C
C
C        LOOP ENTERED TWICE ONCE FOR FLUTTER SPEED MARGIN AND
C        AGAIN FOR FLUTTER Q MARGIN
C        J = 2 IS PASS THROUGH LOOP 500 FOR FLUTTER Q MARGIN
   99 DO 500 K=1,3
C
C        K = 1 WING     K = 2 HORIZONTAL     K = 3 VERTICAL
      GO TO (100,200,300),K
  100 IF(TFH(2)) 500,500,102
  102 DO 110 I=1,38
      TF(I) = TFH(I)
  110 CONTINUE
```

CARD NO **** CONTENTS ****

```
      115 CALL QSUB(N)
          DO 120 I=1,N
          IF(XMISC(K+4) - QQ(I)) 122,120,120
      122 XMISC(K+4) = QQ(I)
          IT(K) = I
          XM(K) = XMATOP(I)
      120 CONTINUE
          GO TO 400
      200 IF(TFH(2)) 500,500,202
      202 DO 210 I=1,38
          TF(I) = TFH(I)
      210 CONTINUE
          GO TO 115
      300 IF(TFV(2)) 500,500,302
      302 DO 310 I=1,38
          TF(I) = TFV(I)
      310 CONTINUE
          GO TO 115
      400 IF(IP(41)) 402,402,500
      402 GO TO (410,420,430),K
      410 WRITE(6,412)
      412 FORMAT(1H0,45X,17HWING FIXED OR AFT)
          GO TO 440
      420 WRITE(6,422)
      422 FORMAT(1H0,46X,15HHORIZONTAL TAIL)
          GO TO 440
      430 WRITE(6,432)
      432 FORMAT(1H0,47X,13HVERTICAL TAIL)
      440 WRITE(6,442)
      442 FORMAT(19X,29HSPEED-ALTITUDE PROFILE POINTS,12X,14HFLUTTER DESIGN/
         * 82X,12HCOMPRESSIBLE/19X,8HALTITUDE,6X,4HMACH,5X,7HDYNAMIC,11X,
         * 4HMACH,5X,7HDYNAMIC,9X,7HDYNAMIC / 21X,4HFEET,7X,6HNUMBER,3X,
         * 8HPRESSURE,10X,6HNUMBER,3X,8HPRESSURE,8X,8HPRESSURE)
          WRITE(6,444) (ALTA(I),XMA(I),QA(I),XMNI(I),Q(I),QQ(I),I=1,N)
      444 FORMAT(16X,F11.0,F11.4,F11.1,5X,F11.4,F11.1,5X,F11.1)
      500 CONTINUE
C
C         END LOOP 500 PASS THROUGH CRITICAL COMPRESSIBLE Q
          IF(J-1) 510,510,600
      510 J = 2
          IF(IP(41)) 512,512,520
      512 WRITE(6,514) FMM
      514 FORMAT(1H1,42X,18HFLUTTER Q MARGIN =,F7.4,22X,
         *    20H** WHVQQ - IP(41) **)
      520 DO 530 I=1,N
          XMNI(I) = XMA(I)
      530 CONTINUE
          GO TO 99
C
C         CHECK FOR VARIABLE SWEEP WING VEHICLE WITH WING IN
C         FORWARD POSITION   STATEMENTS 600 THROUGH 672
      600 IF(TFWF(2)) 700,700,602
      602 IF(IP(41)) 604,604,610
      604 WRITE(6,22)
       22 FORMAT(1H1,88X,20H** WHVQQ - IP(41) **)
          WRITE(6,606)
      606 FORMAT(1H0,48X,12HWING FORWARD)
          WRITE(6,25) FM
       25 FORMAT(1H0,40X,22HFLUTTER SPEED MARGIN =,F5.2)
C
C         N = 3 IS NUMBER OF SPEED ALTITUDE PROFILE POINTS
      610 N = 3
          J = 1
          DO 620 I=1,N
          PO = PRESH(ALTF(I))
          TO = TEMALT(ALTF(I),1.0)
          CALL QINC(XMF(I),PO,TO,QA(I))
          XMNI(I) = FM*XMF(I)
          Q(I) = FMM*QA(I)
      620 CONTINUE
          DO 630 I=1,38
```

CARD NO **** CONTENTS ****

```
          TF(I) = TFHF(I)
      630 CONTINUE
          QQHF = 0.0
          ITH = 0
      640 CALL QSUB(N)
          DO 642 I=1,N
          IF(QQHF - QQ(I)) 644,642,642
      644 QQHF = QQ(I)
          ITH = I
          XMHF = XMATOP(I)
      642 CONTINUE
          IF(J-1) 650,650,670
      650 IF(IP(4)) 652,652,660
      652 WRITE(6,442)
          WRITE(6,444) (ALTF(I),XMF(I),QA(I),XMN(I),Q(I),QQ(I),I=1,N)
      660 J = 2
          DO 662 I=1,N
          XMN(I) = XMF(I)
      662 CONTINUE
          GO TO 640
      670 IF(IP(4)) 672,672,700
      672 WRITE(6,515) FMM
      515 FORMAT(1H0,42X,18HFLUTTER Q MARGIN =,F7.4)
          WRITE(6,442)
          WRITE(6,444) (ALTF(I),XMF(I),QA(I),XMN(I),Q(I),QQ(I),I=1,N)
C
C         * * *  T-TAIL FLUTTER  * * *
C
      700 IF(XMISC(53)) 800,800,702
      702 CALL READMS(1,GJDAT(1),100,37)
          N = 9
          J = 1
          CTT = 0.0
          ITT = 0
          QTT = 0.0
          FACT1 = 0.0
          NP = PC
          DTT = ABS(SPAL(12))
          DO 710 I=1,N
          XMN(I) = FM*XMA(I)
          Q(I) = FMM*QA(I)
      710 CONTINUE
C
C         * * *  START LOOP 760  * * *
C
      712 DO 760 I=1,N
C     FIND CTT VALUE FROM CURVE DATA - - -
C           CTTD1, CTTD2  TWO DIHEDRAL ANGLES FOR WHICH CURVES ARE GIVEN
C
          IFEXT = 1
          IF(XMN(I) - CTTM(1)) 714,724,718
      714 IC = 2
          GO TO 720
      718 DO 718 K=2,NP
          IC = K
          IF(XMN(I) - CTTM(K)) 722,726,718
      718 CONTINUE
      720 IFEXT = 2
C         2 POINT INTERPOLATION OR EXTRAPOLATION.
      722 TMP = (XMN(I) - CTTM(IC-1))/(CTTM(IC) - CTTM(IC-1))
          CTTL = CTT1(IC-1) + (CTT1(IC) - CTT1(IC-1))*TMP
          CTTU = CTT2(IC-1) + (CTT2(IC) - CTT2(IC-1))*TMP
          GO TO 728
C         1 POINT - MACH EQUAL TO CURVE POINT.
      724 IC = 1
      726 CTTL = CTT1(IC)
          CTTU = CTT2(IC)
C     CROSS PLOT
      728 IF(DTT - CTTD1) 738,730,732
      730 TT(I) = CTTL
          GO TO 740
```

```
  732 IF(DTT - CTTD2) 738,734,736
  734 TT(I) = CTTU
      GO TO 740
  736 IFEXT = 2
  738 TT(I) = CTTL + (DTT - CTTD1)/(CTTD2 - CTTD1)*(CTTU - CTTL)
C
  740 IF(IFEXT - 2) 746,742,746
  742 WRITE(6,744) XMN(I),TT(I)
  744   FORMAT( 60H0*****  EXTRAPOLATED ON T-TAIL STIFFNESS COEFF. FOR M
     1ACH NO.,   1F6.2,   8H, CTT = , 1E9.3 )
  746 FACT(I) = Q(I)*TT(I)
      IF(FACT1 - FACT(I)) 740,760,760
  748 CTT = TT(I)
      QTT = Q(I)
      ITT = I
      FACT1 = FACT(I)
      XMTD = XMN(I)
  760 CONTINUE
C
C       * * *  END LOOP 760  * * *
C
      IF(J-1) 762,762,780
  762 IF(IP(4)) 764,764,772
  764 WRITE(6,22)
      WRITE(6,766) SPAL(12)
  766 FORMAT(1H0,37X,14HT-TAIL FLUTTER,5X,10HDIHEDRAL =,F5.1)
      WRITE(6,25) FM
      WRITE(6,768)
  768 FORMAT(11X,29HSPEED-ALTITUDE PROFILE POINTS,12X,14HFLUTTER DESIGN/
     * 11X,8HALTITUDE,8X,4HMACH,5X,7HDYNAMIC,11X,4HMACH,5X,7HDYNAMIC,
     *  9X,3HCTT,12X,5HQ*CTT/13X,4HFEET,7X,6HNUMBER,3X,8HPRESSURE,
     * 10X,6HNUMBER,3X,8HPRESSURE)
      WRITE(6,770)(ALTA(I),XMA(I),QA(I),XMN(I),Q(I),TT(I),FACT(I),
     * I=1,N)
  770 FORMAT(8X,F11.0,F11.4,F11.1,5X,F11.4,F11.1,2E16.8)
  772 J = 2
      DO 774 I=1,N
      XMN(I) = XMA(I)
  774 CONTINUE
      GO TO 712
  780 IF(IP(4)) 782,782,800
  782 WRITE(6,515) FMM
      WRITE(6,768)
      WRITE(6,770)(ALTA(I),XMA(I),QA(I),XMN(I),Q(I),TT(I),FACT(I),
     * I=1,N)
      WRITE(6,784)
  784 FORMAT(1H0,48X,12HDESIGN POINT/)
      WRITE(6,768)
      WRITE(6,770)ALTA(ITT),XMA(ITT),QA(ITT),XMTD,QTT,CTT,FACT1
C
C       CALCULATE SKIN TEMPERATURE AT CRITICAL FLUTTER SPEED AND
C       ALTITUDE
  800 DO 810 I=1,3
      IF(IT(I)) 810,810,812
  812 K = IT(I)
      AL(I) = ALTA(K)
      CALL TEMPER(XM(I),AL(I),PRE,TLOC,TTOT,SFX,TSKR,TSF(I),IER)
  810 CONTINUE
C
C       CALCULATE SKIN TEMPERATURE AT CRITICAL FLUTTER SPEED AND
C       ALTITUDE FOR WING IN FORWARD POSITION ON VARIABLE SWEEP WING
      IF(TFWF(2)) 850,850,820
  820 AWF = ALTF(ITW)
      CALL TEMPER(XMWF,AWF,PRE,TLOC,TTOT,SFX,TSKR,TWF,IER)
C
C
C       CALCULATE TEMPERATURE AT MAXIMUM LIMIT DYNAMIC PRESSURE
C
  850 IF(XMISC(53)) 899,899,852
  852 XMTT = XMA(ITT)
      ALTT = ALTA(ITT)
```

CARD NO **** CONTEN"S ****

```
C       TFTT = SKIN TEMPERATURE AT LIMIT SPEED FOR MAXIMUM Q
      CALL TEMPER(XMTT,ALTT,PRE,TLOC,TTOT,SFX,TSKN,TFTT,IER)
C
C     **** LOOP 900                                          ****
C     **** DETERMINE Q FOR THE WING, HORIZONTAL AND VERTICAL  ****
C     **** (QWING,QHT,QVT) AT THE CONDITION WHICH GIVES  ****
C     **** THE MAXIMUM PRESSURE                              ****
C
  899 QWING = 0.0
      QHT = 0.0
      QVT = 0.0
C
      DO 900 K=1,5
C
C
C       K = 1 WING FIXED OR AFT   K = 2 HORIZONTAL     K = 3 VERTICAL
C       K = 4 WING FORWARD, VARIABLE SWEEP WING
C       K = 5 VERTICAL TAIL FOR T-TAIL
C
      GO TO (901,902,903,904,303),K
C
  901 IF(XMISC(15))900,900,921
C
  921 DO 905 N=1,6
      TEMP(N) = TEMPW(N)
  905 Q(N) = QW(N)
      QTEMP = TSF(1)
      NTEMP = NWING
      GO TO 910
C
  902 IF(XMISC(19))900,900,922
C
  922 DO 906 N=1,6
      TEMP(N) = TEMPH(N)
  906 Q(N) = QH(N)
      QTEMP = TSF(2)
      NTEMP = NHOR
      GO TO 910
C
  903 IF(XMISC(23))900,900,923
C
  923 DO 907 N=1,6
      TEMP(N) = TEMPV(N)
  907 Q(N) = QV(N)
      NTEMP = NVER
      IF(K - 3) 908,908,909
  908 QTEMP = TSF(3)
      GO TO 910
  909 IF(XMISC(53)) 909,900,916
  916 QTEMP = TFTT
      GO TO 910
  904 IF(TFWF(2)) 900,900,931
  931 IF(XMISC(15)) 900,900,932
  932 DO 935 N=1,6
      TEMP(N) = TEMPW(N)
  935 Q(N) = QW(N)
      QTEMP = TWF
      NTEMP = NWING
C
  910 IF(NTEMP - 1)952,952,955
C
  952 QQ = Q(1)
      GO TO 990
C
  955 IF(QTEMP - TEMP(1))960,960,965
C
  960 L1 = 1
      L2 = 2
      GO TO 985
C
  965 IF(QTEMP - TEMP(NTEMP))975,975,980
```

CARD NO **** CONTENTS ****

```
C
  980 L1 = NTEMP - 1
      L2 = NTEMP
      GO TO 985
C
  975 DO 982 N=2,NTEMP
      IF(QTEMP - TEMP(N))978,978,982
  978 L1 = N-1
      L2 = N
      GO TO 985
  982 CONTINUE
C
  985 QQ = Q(L2) - (Q(L2)-Q(L1)) * (TEMP(L2)-QTEMP)/(TEMP(L2)-TEMP(L1))
C
  990 GO TO (911,912,913,914,915),K
C
  911 QWING = QQ
      GO TO 900
C
  912 QHT = QQ
      GO TO 900
C
  913 QVT = QQ
      GO TO 900
C
  914 QHF = QQ
      GO TO 900
. 915 SPAL(10) = QQ
  900 CONTINUE
C
      XMISC(28) = QWING
      XMISC(29) = QHT
      XMISC(30) = QVT
C
C        WRITE SUMMARY FOR CRITICAL FLUTTER WING FIXED OR AFT,
C        HORIZONTAL TAIL, AND VERTICAL TAIL
C
      IF(IP(4)) 992,992,1000
C
  992 IF(TFWF(2))997,997,998
  998 WRITE(6,999)
  999 FORMAT(//////)
      GO TO 1005
  997 WRITE(6,22)
C
 1005 WRITE(6,994)
  994 FORMAT(33X,42H*** DESIGN TEMPERATURE, PRESSURE AND Q *** //18X,
     * 13HPROFILE POINT,4X,8HALTITUDE,3X,8HMACH NO.,3X,11HTEMPERATURE,
     * 3X,8HPRESSURE,4X,7HQ (PSI) )
      WRITE(6,996) (IT(I),AL(I),XM(I),TSF(I),XMISC(I+4),XMISC(I+27),
     * I=1,3)
  996 FORMAT(6X,4HWING,8X,I7,6X,F11.0,F11.4,F14.1,F11.1,F11.0/
     * 6X,10HHORIZONTAL,2X,I7,6X,F11.0,F11.4,F14.1,F11.1,F11.0/
     * 6X,8HVERTICAL,4X,I7,6X,F11.0,F11.4,F14.1,F11.1,F11.0)
C
C
 1000 CALL WRITMS(1,SPAL(1),50,38)
C
C
C
      RETURN
      END
```

Section V

## REFERENCES


1. Spalding, D. B., and Chi, S. W., "The Drag of a Compressible Turbulent Boundary Layer on a Smooth Flat Plate With and Without Heat Transfer." Journal of Fluid Mechanics, January 1964, p 117

2. Hopkins, E. J., and Inouye, M., "An Evaluation of Theories for Predicting Heat Transfer on Flat Plates at Supersonic and Hypersonic Mach Numbers," AIAA Journal, Vol 9, No. 6, June 1971, p 993

3. Hilsenrath, J., and Klein, M., of National Bureau of Standards, "Tables of Thermodynamic Properties of Air in Chemical Equilibrium Including Second Virial Corrections from 1500°K to 15,000°K," Arnold Engineering Development Center Reports AEDC-TDR-65-58, March 1965

4. Truitt, R. W., Fundamentals of Aerodynamic Heating, Library of Congress Catalog No. 629.13237 T, The Ronald Press Company, New York, 1960

5. Eckert, R. R. G., "Survey on Heat Transfer at High Speeds," Wright Aeronautical Development Center TR-54-70, April 1954

6. Van Driest, E. E., "The Turbulent Boundary Layer for Compressible Fluids on a Flat Plate with Heat Transfer," North American Aviation Report Al-997, 27 January 1950

7. Van Driest, E. R., "The Turbulent Boundary Layer with Heat Transfer," 50 Jahre Grenzschiftforschung, Verlag Friedr, Vieweg & Sohn, Braunschweig, Germany, p 257

8. Donaldson, Coleman du Pont, "Note on the Importance of Imperfect Gas Effects and Variation of Heat Capacities on the Isentropic Flow of Gases," NACA Document RM No. L8J14, 10 December 1948

9. Liepmann, H. W., and Roshko, A., Elements of Gas Dynamics, Library of Congress Catalog No. 533.13, John Wiley & Sons, New York, 1957

10. Bloom, M., "Thermal Conditions Associated with Aircraft in Flight," WADC TR 55-169, March 1956

11. United States Committee on Extension of the Standard Atmosphere, U.S. Standard Atmosphere, 1962, Library of Congress Catalog No. 551. U.S. Government Printing Office, Washington, D.C., December 1962

12. Anonymous, "Climatic Extremes for Military Equipment," Military Specification MIL-STD-210A, revised 30 November 1958

13. Anonymous, "Transparent Areas, Anti-icing, Defrosting and Defogging Systems, General Specification for," Military Specification MIL-T-5842A, amended 13 July 1953

14. Torgeson, W. L., Johnson, J. C., and Wright, N. B., "An Engineering Study of Air Conditioning Load Requirements for Aircraft Compartments," WADC TR 55-254, June 1955

15. Boelter, L. M. K., Poppendiek, H., Young, G., and Andersen, J. R., "An Investigation of Aircraft Heaters XXX - Nocturnal Irradiation as a Function of Altitude and Its Use in Determination of Heat Requirements of Aircraft," NACA TN 1454, January 1949

16. Haroldson, H., "A Function Subprogram for Solving 4th Power Equations," North American Aviation Report NA-65-948, 6 December 1965

17. Scanlan and Rosenbaum, Introduction to the Study of Aircraft Vibration and Flutter, the MacMillan Company, New York, 1951

18. Roberts, William H., "A Stiffness Criterion for Flutter and the Optimum Spanwise Skin Distribution for Aeroelastic Problems," NA-54-619, Rockwell International, Los Angeles, Calif., 1954

19. Land, Norman S., and Fox, Annie G., "An Experimental Investigation of the Effect of Mach Number, Stabilizer Dihedral, and Fain Torsional Stiffness on the Transonic Flutter Characteristics of a Tee-Tail," TND-924, National Aeronautics and Space Administration, Virginia, 1957

20. Hodson, C.H., "Stiffness Requirements to Prevent Flutter of Moderate to High Aspect Ratio Surfaces." NA-64-745, Rockwell International, Los Angeles, Calif., 1964

21. Posluszny, D.A., and Carpenter, J., "Derivation of A Spanwise Torsional Constant J, for a "Multi-Station Weight Increment," NA-61-1039, Rockwell International, Los Angeles, Calif., 1961

PART 3

FATIGUE MODULE

Section 1

# INTRODUCTION AND SUMMARY

## PROGRAM OBJECTIVES

The objective of the fatigue module is to provide a means for evaluating fatigue requirements commensurate with the preliminary design definitions of structural components. In this phase of vehicle synthesis, it is not practical to examine all of the structural elements. Therefore, the scope of this module has been limited to the evaluation of components which would most probably be designed by fatigue considerations. These components are the wing lower cover subjected to bending loads and fuselage panels subjected to cyclic pressurization.

## APPROACH TO FATIGUE EVALUATION

Fatigue life prediction by the method of "strain cycling" has been selected for use in the structural weight estimation program.

Strain cycling evaluates fatigue damage in two categories, or blocks, corresponding to material fatigue behavior observed under cyclic loading. After an unnotched specimen has been cycled between fixed fully reversed strain limits to approximately 20 percent of its fatigue life, the material stabilizes and no further change will take place. These stabilized values of stress and strain more closely represent material behavior than conventional stress-strain diagrams. Stabilized values are plotted in a "cyclic" stress-strain curve. In addition, a curve relating strain amplitude to fatigue life is derived from experimental data. These curves are used to derive fundamental equations, which are summarized in this report.

The stress level at the edge of a notch may often be greater than the yield stress, but the preponderant elastic region forces the limited yield region to be strain controlled. Some tests have shown that failure of the notched specimen is precipitated at the edge of the notch in the same number of cycles as an unnotched specimen subjected to the same strain range. Using equivalent unnotched strain, in conjunction with unnotched strain-cycling material curves, the life of a notched specimen can be predicted for a wide range of strain amplitudes. This range includes the case of fully reversed cycles ($R = -1$). The strain amplitude of the notch is determined from the hysterisis curves, which are defined in this report. Then, by assuming linearity between cyclic strain amplitude and cyclic mean stress for a given life in the modified Goodman Diagram, the equivalent strain at edge of notch is obtained for $R = -1$.

The residual stress used in the second and subsequent blocks is that residual generated by the largest strain range in the first block. The differences between this residual stress and residual stress generated for each level are added to the mean stress for each level in the second and subsequent blocks. This mean stress is next added to stress amplitude to determine the new maximum and minimum stresses. Finally, the life for each load level is determined using the modified Goodman diagram and Miner's rule.

## SUMMARY OF ANALYSIS CAPABILITIES

The strain cycling approach is used to obtain allowable design tensile stress for the wing lower cover and for fuselage components. Material property, spectrum, and construction data are used to obtain this fatigue "cutoff stress." Material property and construction data required by this program are:

- Ultimate tensile strength, psi
- Modulus of elasticity, psi
- Reduction of area, RA
- Elastic stress concentration factor, $K_t$

Required criteria data consist of:

- Spectrum data
- Reference design load
- Service life
- Scatter factor

Wing bending moment spectra are evaluated at two wing stations. Bending moments are correlated to nominal stresses for which life calculations are performed. An iteration procedure is used to solve for nominal spectrum stresses that meet the required vehicle life. Correlation between strength design bending moment and the spectrum bending moments provides an allowable tensile stress at the reference loading.

Fuselage cover and minor frame material fatigue evaluation is performed for pressure cycling and material endurance limit. Spectrum pressures are correlated to nominal stresses, and calculations for life are performed. Assuming that the design pressure is the maximum pressure defined by the spectrum, the allowable tensile stress is also the maximum nominal stress that satisfies required vehicle life.

Endurance limit is required for fuselage acoustic fatigue evaluation. The calculations are performed for:

- Elastic stress concentration factor, $K_t$, equal to one (polished specimen)
- Fully reversing load ($R = -1$)
- Infinite number of cycles ($n = 10^9$)

## MODULE STRUCTURE AND OPERATION

This program is written in FORTRAN IV extended programming language for operation on the CDC 6600 computer and is structured in a single overlay within 50,000 octal core locations. This module may be executed as a stand-alone program, or as part of the integrated structural weight estimation program. In the stand-alone mode, wing bending moment spectra, fuselage pressure spectrum, and component material identification number are required input. In the integrated mode, wing bending moment spectra is provided by the airload module.

Program printed output is controlled by user specifications. The output consists of input data tables and intermediate steps in the life calculations. Error messages are printed, should certain convergence problems occur. These messages contain pertinent data and description of the calculation process involved.

## Section II

## METHODS AND FORMULATIONS

### GENERAL DISCUSSION OF METHODOLOGY

The analytical foundation for the strain cycling method has been obtained from references 1 through 11. This report discusses some of the background, but is basically oriented toward the discussion of actual programmed equations and solutions. This manual discusses the strain cycling method and its application to vehicle spectrum loading evaluation and life prediction.

### FATIGUE FORMULATIONS

#### CYCLIC STRESS-STRAIN RELATIONSHIPS

The basic equations of all stress analysis are the equilibrium equations involving stresses and the compatibility equations involving total strains. The relationship between stress range and total strain range will suffice in many applications involving cyclic straining, but it is desirable to separate the total strain into elastic and plastic components (equation 1) and to express each of these components in terms of cyclic life.

$$\epsilon_t = \epsilon_{e\ell} + \epsilon_p \qquad (1)$$

If a plot is made on logarithmic coordinates of the strain range versus the number of cycles to failure, the result is found to be very nearly two straight lines, as shown in Figure 30.

The cyclic life, $N_f$, is related to the plastic strain per cycle, $\epsilon_p$, by a power law (equation 2).

$$\epsilon_p = C_2 N_f^{-\gamma} \qquad (2)$$

where $C_2$ and $\gamma$ are material constants.

If the stress range at the half life is divided by the elastic modulus, the quotient can be regarded as the elastic-strain range associated with the cyclic life $N_f$. A plot of this strain versus cyclic life on log.-log.

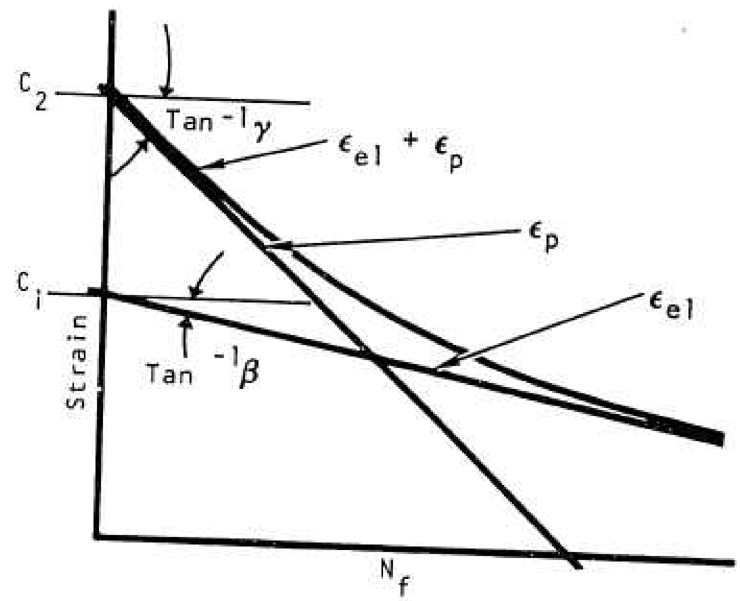



Figure 30. Strain range versus number of cycles to failure.

coordinates also results in a reasonably straight line (Figure 30). This elastic strain, $\epsilon_{e\ell}$, can also be defined by a power law, as shown in equation 3.

$$\epsilon_{e\ell} = C_1 N_f^{-\beta} \tag{3}$$

where $C_1$ and $\beta$ are other material constants.

By definition, elastic strain is proportional to stress (equation 4) such that cyclic life can be defined as shown in equation 5.

$$\epsilon_{e\ell} = \frac{\sigma}{E} \tag{4}$$

$$N_f = \left(\frac{C_1 E}{\sigma}\right)^{1/\beta} \tag{5}$$

The total strain can then be defined by equation 6 or 7.

$$\varepsilon_t = C_1 N_f^{-\beta} + C_2 N_f^{-\gamma} \qquad (6)$$

$$\varepsilon_t = \frac{\sigma}{E} + C_2\left(\frac{\sigma}{C_1 E}\right)^{\gamma/\beta} \qquad (7)$$

Material constants $C_1$, $C_2$, $\beta$, and $\gamma$ are not readily available for most structural materials. However, these constants can be approximated from empirical relationships which use material properties obtained from tensile tests. Material properties that are determined from tensile tests are E, $F_{tu}$, and RA.

Where

$F_{tu}$ = ultimate tensile strength, psi

RA = reduction in area as defined by equation 8

$$RA = \frac{A_o - A_f}{A_o} \qquad (8)$$

Where

$A_o$ = initial specimen cross-section area

$A_f$ = final fracture cross-section area

Ductility, D, is defined as a logarithmic value based on measurement of area (equation 9).

$$D = \ln\left(\frac{A_o}{A_f}\right) = \ln\left(\frac{1}{1-RA}\right) \qquad (9)$$

Fracture stress, $\sigma_f$, is determined from equation 10.

$$\sigma_f = F_{tu}\ (1+D) \tag{10}$$

Equations 11 through 14 can then be used to obtain approximations of the material constants that define the cyclic stress-strain curve.

$$\beta = 0.0792 + 0.179\ \log.\left(\frac{\sigma_f}{F_{tu}}\right) \tag{11}$$

$$\gamma = \log.\left[\frac{3.31\ \sqrt[4]{D}}{\sqrt[3]{1 - 81.4\ \frac{F_{tu}}{E}\left(\frac{\sigma_f}{F_{tu}}\right)^{0.179}}}\right] \tag{12}$$

$$C_1 = 1.12\ \frac{F_{tu}}{E}\left(\frac{\sigma_f}{F_{tu}}\right)^{0.893} \tag{13}$$

$$C_2 = 0.125\ (D)^{0.75}\ (10)^{\gamma} \tag{14}$$

STRESS AND STRAIN AT EDGE OF NOTCH

The cyclic stress at the edge of the notch, $\sigma_{max}$, is obtained by Neuber's rule (equation 15).

$$E\sigma\varepsilon = \left(K_t f_{max}\right)^2 \tag{15}$$

Where

$K_t$ = elastic stress concentration factor

$f_{max}$ = nominal net section maximum stress for a cycle, psi

If strain at the notch (equation 7) is substituted in equation 15, the maximum stress can be represented by equation 16.

$$E\sigma_{max}\left[\frac{\sigma_{max}}{E} + C_2\left(\frac{\sigma_{max}}{C_1 E}\right)^{\gamma/\beta}\right] = (K_t f_{max})^2 \tag{16}$$

Equation 17 is another form of equation 16 which can be solved for stress by successive iterations.

$$\sigma_{max}^2 + \frac{C_2 E}{(C_1 E)^{\gamma/\beta}}\sigma_{max}^{(\gamma/\beta + 1)} - (K_t f_{max})^2 = 0 \tag{17}$$

For this maximum stress at the edge of the notch, the maximum strain, $\epsilon_{max}$, is calculated from equation 7.

## HYSTERESIS CURVES

Hysteresis curves are dependent on the relationship between the nominal net section maximum and minimum stress for a cycle. The two unique cases are depicted in Figures 31 and 32.

Points on the hysteresis curves are defined for any stress ratio and effective stress concentration factor $K_n$. Neuber's formula (equation 18) defines $K_n$ in terms of the elastic stress concentration factor, $K_t$, and notch geometry.

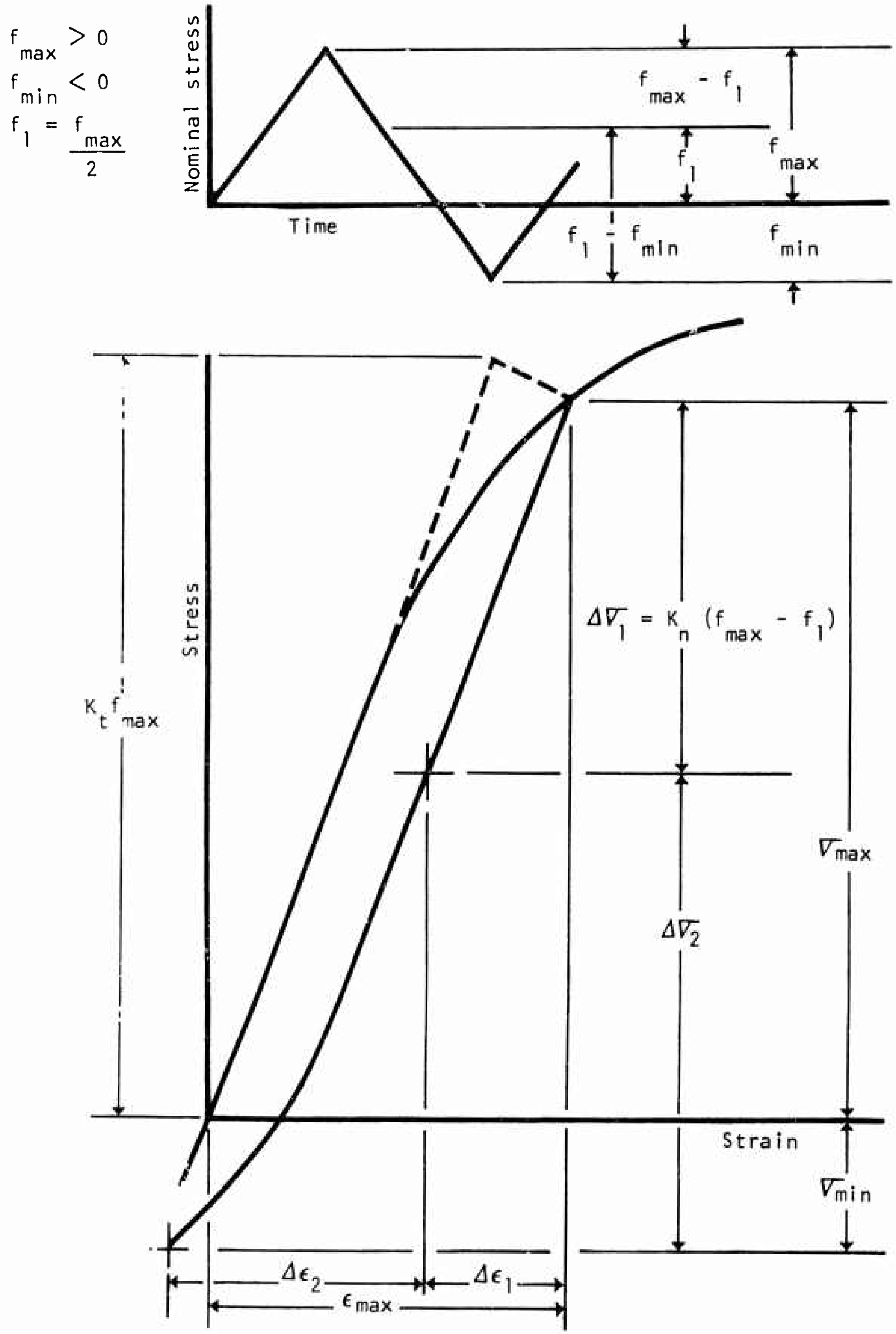



Figure 31. Hysteresis curve for $f_{min}$ less than zero.

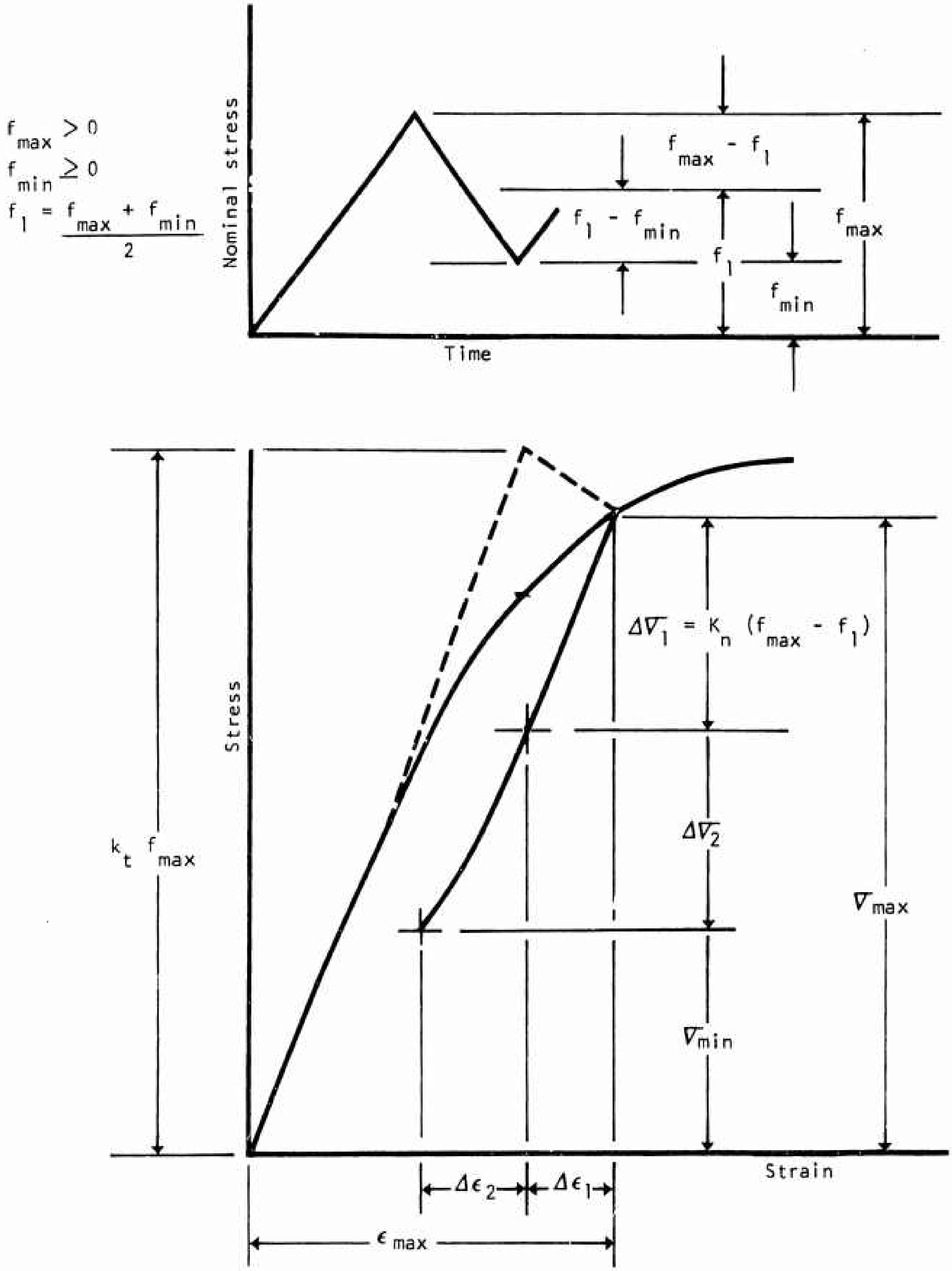



Figure 32. Hysteresis curve for $f_{min}$ greater than zero.

$$K_n = 1 + \frac{K_t - 1}{1 + \frac{\pi}{\pi - w}\sqrt{\frac{A}{r}}} \tag{18}$$

Where

$w$ = notch angle, radians

$r$ = notch radius, inches

$A$ = material constant, inches (Neuber parameter)

For the purpose of this program, which is intended to evaluate "average" stress concentration rather than a specific notch, $K_n$ is assumed to be equal to $K_t$. This approximation can be made, since the Neuber parameter, A, is a small number on the order of 0.005. For a given $K_t$ this assumption leads to a slightly conservative fatigue life evaluation.

Unloading along the cyclic stress-strain curve is approximated by considering two segment: (1) an elastic segment $\Delta\sigma_1$, and (2) an elastoplastic segment $\Delta\sigma_2$ due to nonlinearity of the stress-strain curve, as shown by equation 7. Stress and strain for the elastic segment are calculated by equations 19 and 20.

$$\Delta\sigma_1 = K_n (f_{max} - f_1) \approx K_t (f_{max} - f_1) \tag{19}$$

$$\Delta\epsilon_1 = \frac{\Delta\sigma_1}{E} \tag{20}$$

The elastoplastic segment of the hysteresis curve is assumed to be proportional to the loading curve, as shown by equation 21.

$$\frac{K_n (f_1 - f_{min})}{\Delta\epsilon_2} = \frac{K_n f_{max}}{\epsilon_{max}} \tag{21}$$

Then

$$\Delta\epsilon_1 = \frac{(f_1 - f_{min})\, \epsilon_{max}}{f_{max}} \tag{21a}$$

and $\Delta\sigma_2$ is obtained by substituting $\Delta\varepsilon_2$ in the cyclic stress-strain equation and solving for stress (equation 7).

Then

$$\sigma_{min} = \sigma_{max} - \Delta\sigma_1 - \Delta\sigma_2 \tag{22}$$

Strain amplitude, $\varepsilon_a$, and mean stress, $\sigma_{mean}$, at the notch for a cycle of maximum-to-minimum nominal stress can then be calculated (equations 23 and 24)

$$\varepsilon_a = \frac{\Delta\varepsilon_1 + \Delta\varepsilon_2}{2} \tag{23}$$

$$\sigma_{mean} = \frac{\sigma_{max} + \sigma_{min}}{2} \tag{24}$$

Every level of load in a spectrum produces a residual stress. Residual stress is that stress at the notch corresponding to a zero nominal stress. Residual stress is dependent on the stress ratio, as shown in Figures 31 and 32.

For the loading cycle shown in Figure 31, where:

$$f_{max} > 0$$

$$f_{min} < 0$$

$$\sigma_{Res} = \sigma_{min} - \frac{K_n f_{min}}{2} - \Delta\sigma_{Res2} \tag{25}$$

Where $\Delta\sigma_{Res2}$ corresponds to $\Delta\varepsilon_{Res2}$, which is determined from equation 26.

$$\Delta\varepsilon_{Res2} = \frac{f_{min}\ \varepsilon_{max}}{2\ f_{max}} \tag{26}$$

Then, $\Delta\sigma_{Res2}$ is obtained by an approximation-iteration solution of equation 7.

For the loading cycle shown in Figure 32, where:

$$\varepsilon_{max} > 0$$

$$\varepsilon_{min} \geq 0$$

$$\sigma_{RES} = \sigma_{max} - \frac{K_n \varepsilon_{max}}{2} - \Delta\sigma_{Res2} \tag{27}$$

Where $\Delta\sigma_{Res2}$ corresponds to $\Delta\varepsilon_{Res2}$, which is determined from equation 28.

$$\Delta\varepsilon_{Res2} = \frac{\varepsilon_{max}}{2} \tag{28}$$

Then, $\Delta\sigma_{Res2}$ is obtained by solving equation 7.

LIFE CALCULATION

Figure 33 shows a modified Goodman diagram assuming linearity between cyclic strain amplitude and cyclic mean stress for a given life. From this relationship, the equivalent strain, $\varepsilon_T$, at the edge of a notch for fully reversed loading (R = -1) can be determined (equation 29).

$$\varepsilon_T = \frac{\varepsilon_a}{1 - \frac{\sigma_{mean}}{\sigma_f}} \tag{29}$$

The number of cycles to failure, $N_f$, corresponding to $\varepsilon_T$ is then obtained by an approximation-iteration approach using equation 6. Cycles to failure calculated for the spectrum loading by this approach constitute the first block.

Each of the load cycle levels in a spectrum generates a residual stress (equation 25 or 27). The residual stress generated by the largest strain

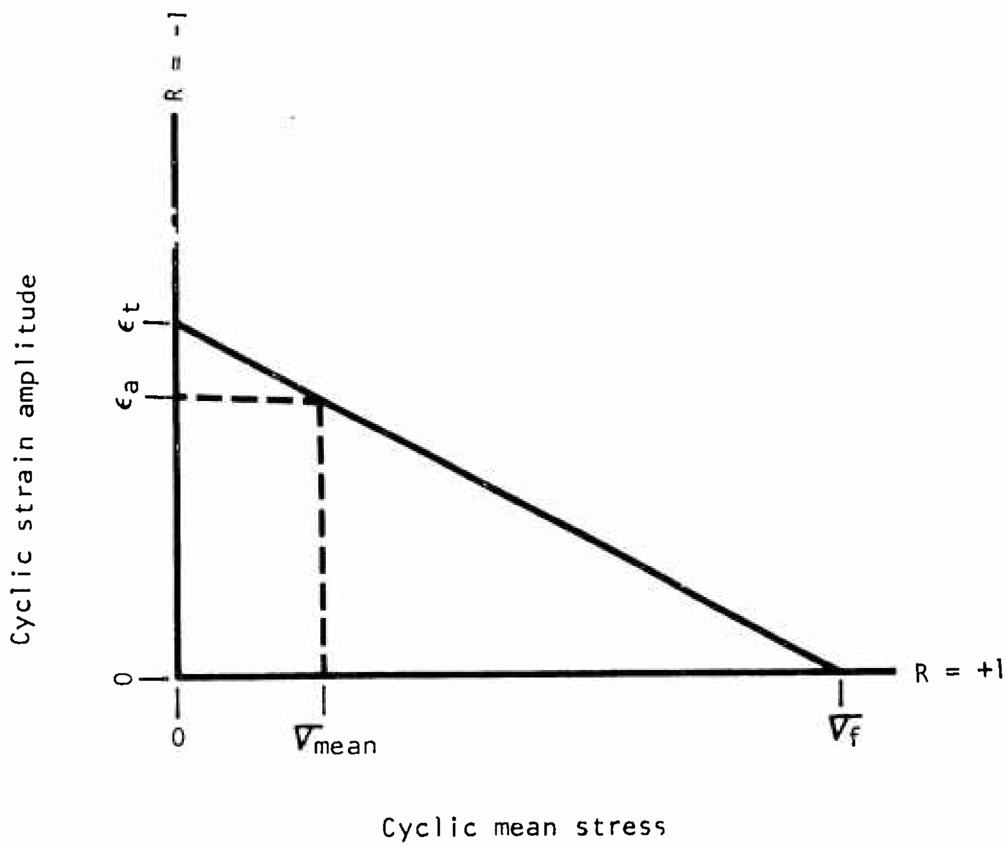



Figure 33. Modified Goodman diagram.

range $(\Delta\epsilon_1 + \Delta\epsilon_2)$ in the first block is used in the second block. The difference between this residual stress, $\sigma^*_{RES}$, and the residuals calculated for each of the load cycle levels is used to determine the mean stress for the second block (equation 30).

$$\sigma_{mean2} = \sigma_{mean} + \sigma^*_{RES} - \sigma_{RES} \qquad (30)$$

Equivalent strain for the second block is then calculated by equation 31, and cycles to failure are calculated in the same manner as used in the first block.

$$\epsilon_{T_2} = \frac{\epsilon_a}{1 - \frac{\sigma_{mean_2}}{\sigma_f}} \qquad (31)$$

The damage ratio, the ratio of applied cycles to the number of cycles to failure, is then calculated and the total life is obtained by using Miner's cumulative damage rule (equation 32).

$$\sum\left(\frac{n}{N_f}\right)_1 + B\sum\left(\frac{n}{N_f}\right)_2 = 1 \qquad (32)$$

Where

B = number of blocks to failure after first block of loading

Scatter factor, SF, in fatigue evaluation is analogous to safety factor in strength design. Equation 32 represents one lifetime of the structural member (SF = 1) such that the cumulative damage rule, including scatter factor, is shown in equation 33.

$$\sum\left(\frac{n}{N_f}\right)_1 + B\sum\left(\frac{n}{N_f}\right)_2 = \frac{1}{SF} \qquad (33)$$

Solving equation 33 for B,

$$B = \frac{\frac{1}{SF} - \sum\left(\frac{n}{N_f}\right)_1}{\sum\left(\frac{n}{N_f}\right)_2} \tag{34}$$

Life in hours is then calculated by equation 35.

$$\text{Life (SF)} = (1 + B)\, H \tag{35}$$

Where

H = number of hours in each block associated with applied cycles, n, in each block

DESIGN TO LIFE

Material ultimate tensile strength is an allowable or not-to-exceed stress level in static strength analysis. If vehicle service life requirements could also be represented by not-to-exceed or fatigue "cutoff" stress, design to life can be incorporated into the strength sizing procedure. The strain cycle analysis method provides an approach for deriving this cutoff stress from spectrum and material property data. The following formulations present programmed approaches used to evaluate different forms of wing bending moment spectrum and fuselage pressure cycle data.

WING SPECTRA

Wing bending moment-exceedances spectra are provided by the airload module or through user input to this module. Blocked spectra are provided for eight flight segments, two taxi segments, and a ground-air-ground segment. Net bending moments are provided at the wing side of the fuselage station and at an outboard wing station. Moments at the side of the fuselage station are in the fuselage reference system, and moments at the outboard station are in the wing structural reference system. Table 19 shows bending moments and associated gust and maneuver exceedances for a typical flight segment. Table 20 shows typical ground-air-ground data.

TABLE 19. TYPICAL FLIGHT SEGMENT SPECTRA

Spectra Segment No. 7

| Point | SOF Bend MOM | WOS Bend MOM | Exceedances - Gust | Exceedances - Man. | Load Factor |
|---|---|---|---|---|---|
| 1 | 101856512.0 | 93595488.0 | 0.12013E-08 | 0.0 | 3.8 |
| 2 | 93815232.0 | 86206400.0 | 0.20802E-07 | 0.0 | 3.5 |
| 3 | 85773920.0 | 78817264.0 | 0.36023E-06 | 0.0 | 3.2 |
| 4 | 77732608.0 | 71428144.0 | 0.62380E-05 | 0.0 | 2.9 |
| 5 | 69691296.0 | 64039024.0 | 0.10802E-03 | 0.81000E-03 | 2.6 |
| 6 | 61649984.0 | 56649888.0 | 0.18706E-02 | 0.13500E-01 | 2.3 |
| 7 | 53608704.0 | 49260800.0 | 0.32394E-01 | 0.27000E 00 | 2.0 |
| 8 | 45567392.0 | 41871664.0 | 0.56168E 00 | 0.67500E 01 | 1.7 |
| 9 | 37526080.0 | 34482544.0 | 0.11690E 02 | 0.22275E 03 | 1.4 |
| 10 | 29484768.0 | 27093424.0 | 0.55710E 04 | 0.81000E 04 | 1.1 |
| 11 | 24123904.0 | 22167344.0 | 0.55709E 04 | 0.51840E 03 | 0.9 |
| 12 | 16082610.0 | 14778239.0 | 0.11690E 02 | 0.59400E 01 | 0.6 |
| 13 | 8041304.0 | 7389118.0 | 0.56168E 00 | 0.40500E-01 | 0.3 |
| 14 | 0.0 | 0.0 | 0.32394E-01 | 0.27000E-03 | 0.0 |
| 15 | -8041304.0 | -7389118.0 | 0.18706E-02 | 0.0 | -0.3 |
| 16 | -16082610.0 | -14778239.0 | 0.10802E-03 | 0.0 | -0.6 |
| 17 | -24123904.0 | -22167344.0 | 0.62380E-05 | 0.0 | -0.9 |
| 18 | -32165216.0 | -29556464.0 | 0.36023E-06 | 0.0 | -1.2 |
| 19 | -40206528.0 | -36945600.0 | 0.20802E-07 | 0.0 | -1.5 |
| 20 | -48247808.0 | -44334688.0 | 0.12013E-08 | 0.0 | -1.8 |

1G SOF M = 26804352.0
1G WOS M = 24630400.0

TABLE 20. TYPICAL GROUND-AIR-GROUND SEGMENT SPECTRA

| Spectra Segment No. 11 - Ground-Air-Ground Occurrences | | |
|---|---|---|
| SOF Bend MOM | WOS Bend MOM | Occurrences |
| 33515408.0 | 31016528.0 | 0.20000E 04 |
| -34969344.0 | -27504432.0 | 0.20000E 04 |

Flight spectra load factors, Table 19, are preselected such that symmetry about a reference 1g condition is maintained. This preselection of spectra load factors results in symmetry such that gust exceedances for the second 10 points in Table 19 are mirror images of the first 10 points. Figure 34 shows a typical plot of bending moments versus gust exceedances. The upper curve is obtained by plotting points 1 through 10, and the lower curve from points 11 through 20.

Occurrence data (cycles) are obtained by blocking this diagram as depicted by the region bounded by points 3, 4, 17, and 18. Occurrences for this portion of the spectra are calculated from equation 36.

$$n_3 = Ex_4 - Ex_3 = Ex_{17} - Ex_{18} \tag{36}$$

where

$n$ = number of applied cycles for the specific range of bending moments

$Ex$ = exceedances associated with each bending moment

Average maximum and minimum bending for this portion of the spectra (region 3-4-17-18 in Figure 34) are calculated by equations 37 and 38, respectively.

$$M_{max3} = M_4 + \frac{M_3 - M_4}{3} \tag{37}$$

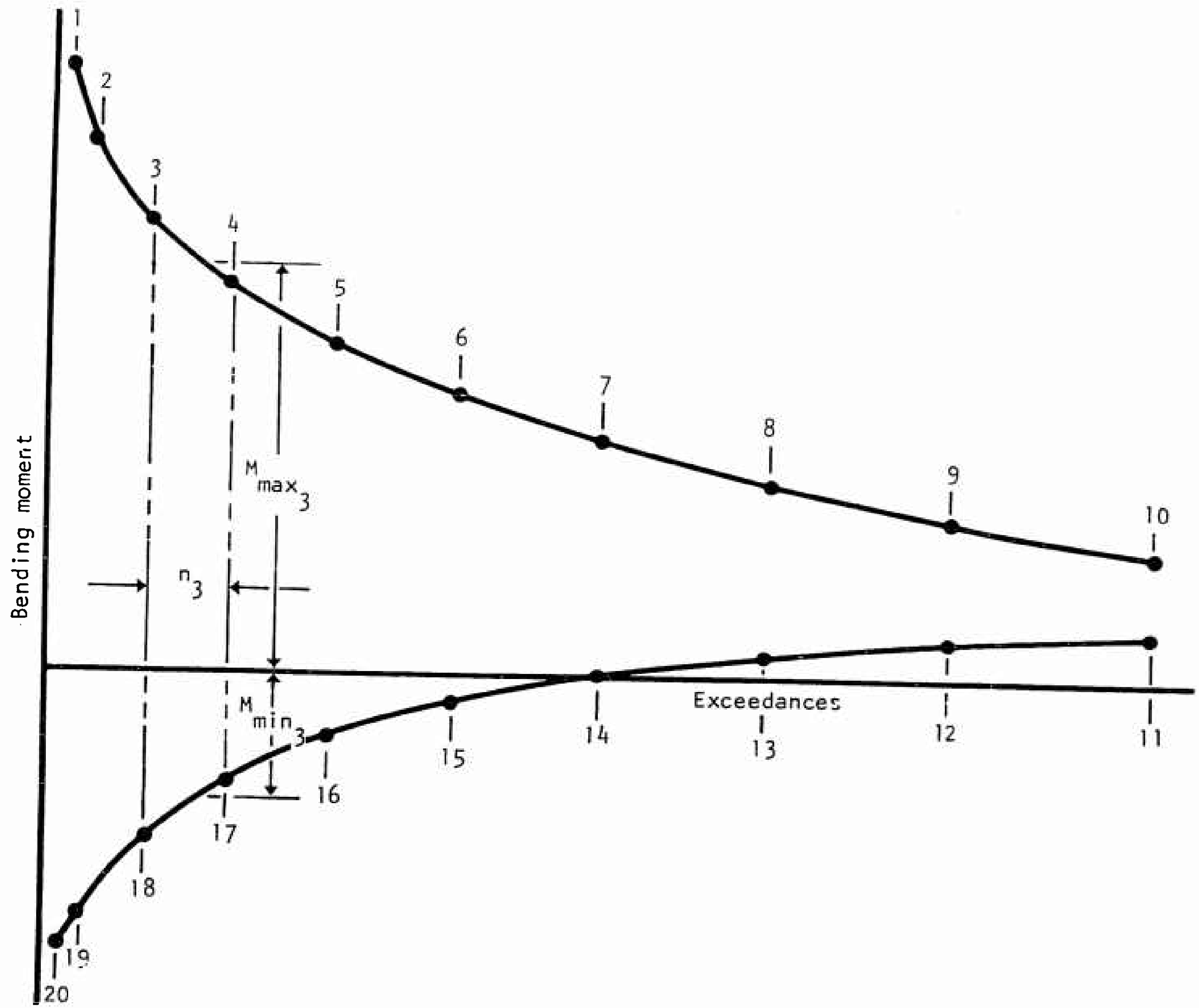



Figure 34. Wing bending moment versus gust exceedances.

$$M_{min3} = M_{17} + \frac{M_{18} - M_{17}}{3} \tag{38}$$

These equations give weighted average moment values which account for curvature of the bending moment exceedance plot. Nine moment-gust occurrence sets are calculated for each of the eight flight segments.

Vehicle maneuvers are assumed to start from a balanced 1g condition such that 1g loads describe one limit of the load range. Figure 35 depicts the first 10 points of a flight segment. Occurrences for each block of this diagram are calculated by equation 36. Should this calculation result in zero occurrences (refer to Table 19), the load range block does not exist and, therefore, is not used in the analysis. Average bending moment for one part of the load range is calculated by either equation 37 or 38, and the other moment is obtained from the 1g condition. The greater of these two moments is the maximum, and the other value is the minimum moment. Eighteen, or less if occurrence is zero, moment-maneuver occurrence sets are calculated for each of the eight flight segments.

Taxi spectra bending moment-occurence values are calculated in the same manner as those used for gust maneuvers.

Ground-air-ground data are in the form of maximum and minimum bending moments and occurrences, which do not require any additional computations.

Load-occurrence data from the flight, taxi, and ground-air-ground segments constitute the total spectrum. As previously discussed, zero occurrence sets are not used. Except for possible residual stress effects, compression-compression cycles have no impact on fatigue life. Therefore, should the maximum bending moment be negative (compression in the lower cover), those cyclic load sets are not used in the fatigue calculations.

Conversion of Bending Moments to Nominal Stress

Axial load in the lower cover can be approximated by equation 39. The assumption is that all of the structure which resists bending loads is concentrated in the cover panels.

$$F = \frac{M}{H} \tag{39}$$

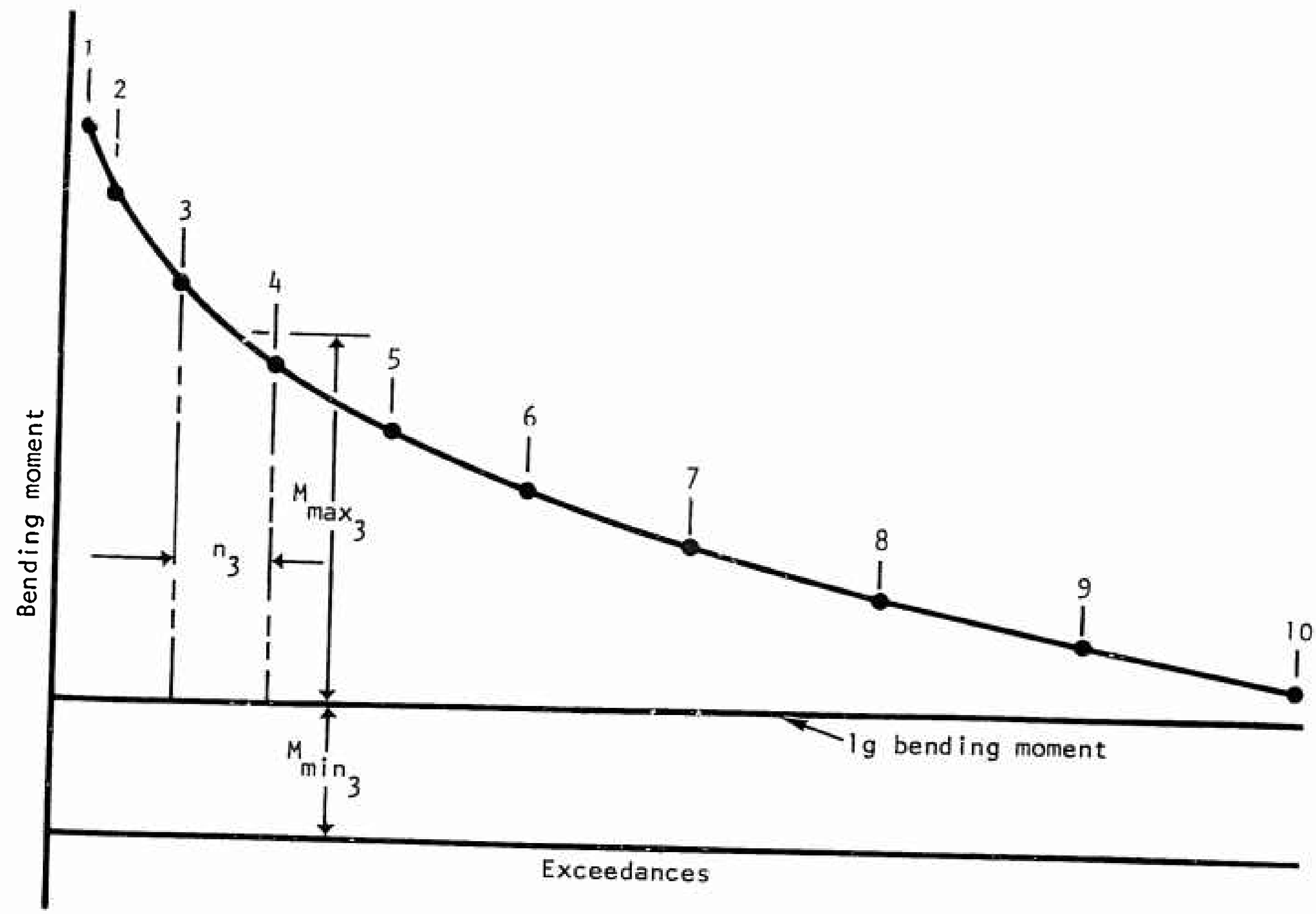



Figure 35. Wing bending moment versus maneuver exceedances.

where

$F$ = Axial load, lb

$H$ = average wing structural box depth, in.

If the area (thickness times width) of the cover panel is known, the stress can then be determined by equation 40.

$$f = \frac{M}{HA} \tag{40}$$

where

$A$ = lower cover cross-section area, in.$^2$

$f$ = average lower cover nominal stress, psi

Both wing depth and section area are fixed quantities for the final sized structure such that stress is directly proportional to bending moment.

$$f = KM \tag{41}$$

where

$K$ = proportionality constant

Application of equation 41 to the average spectrum moments provides the nominal stress data required for life calculation.

Block Size

In general, spectrum block sizes for fatigue calculations are determined for one flight; another rule is to set the block size at a fraction of vehicle service life. The latter approach is used in this program. This fraction (FL) is part of the user input data set. Each of the applied cycles in the

spectrum is multiplied by FL to obtain the appropriate load-cycle data per block. The number of hours (H) associated with these cycles is obtained by multiplying vehicle life by FL.

Determination of Fatigue Cutoff Stress

For a given value of the proportionality constant, K, life can be calculated. If the summation of nominal net section maximum stress versus life is plotted on log.-log. coordinates, the result is very nearly a straight line, as shown in Figure 36. Equation 42 defines the slope of this curve.

$$m = \frac{\log.\ (K_1 \Sigma\ M_{max}) - \log.\ (K_2 \Sigma\ M_{max})}{\log.\ L_2 - \log.\ L_1} \tag{42}$$

If the slope of the curve, m, is known, the proportionality constant that satisfies the required life can be solved by a single life calculation.

$$K_2 = K_1\ e^{m(\log.\ L_2 - \log.\ L_1)} \tag{43}$$

For a given problem, both $K_2$ and m are unknown. Initial estimates $K_o$ and $m_o$ are used in a successive iteration solution.

$$K_o = \frac{F_{tu}}{2M_o} \tag{44}$$

$$m_o = -0.1666667 \tag{45}$$

where

$K_o$ = initial estimate of the proportionality constant

$M_o$ = the largest average bending moment for the spectrum

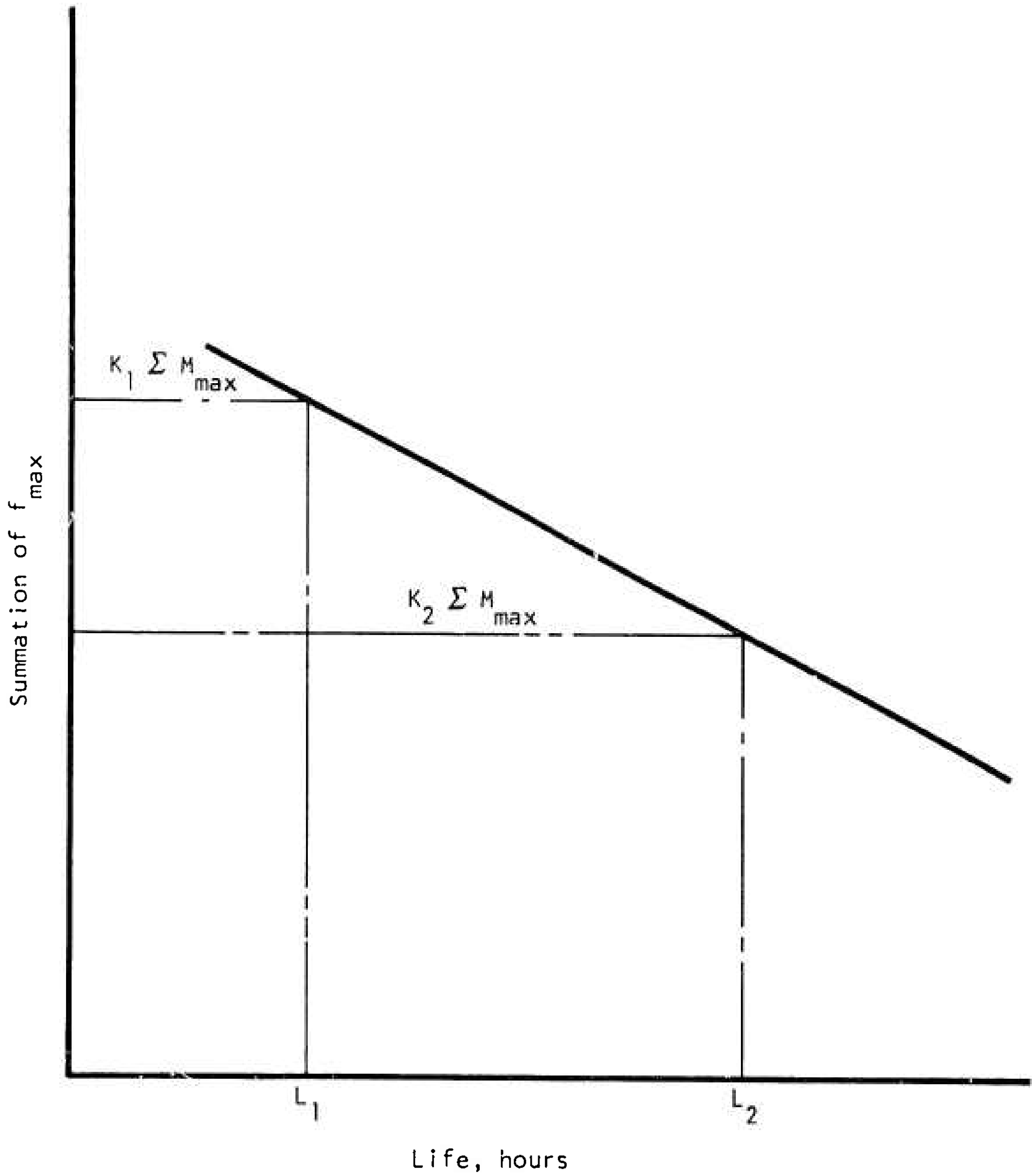



Figure 36. Summation of nominal net section stress versus life.

The initial $K_o$ is used to calculate nominal stresses and life by the strain cycle method. By using the estimated slope $m_o$ in equation 43, $K_2$ can be calculated for the required life $L_2$. However, life calculated by using $K_2$ in the strain cycle solution will not be the required life. The K and L at the two points are used to calculate a new slope estimate (equation 42). The point closest to the required life is then used to repeat the calculation of K and life until the calculated life is within a tolerance (1 percent) of the required life.

As previously defined in equation 41, stress is proportional to bending moment. If the static strength design bending moment is known, the not-to-exceed or cutoff stress at this load, commensurate with the required life, can be calculated by equation 46.

$$f_{allow} = K_2 M_{Ref} \tag{46}$$

where

$f_{allow}$ = cutoff or not-to-exceed stress to satisfy vehicle life requirement

$M_{Ref}$ = static limit strength design bending moment

The foregoing procedure is used to determine the cutoff stress at the two wing stations. Both allowables are stored in the material files as fractions of ultimate tensile strength (equation 47) for access by the wing and empennage weight-estimating module.

$$K_{F_{tu}} = \frac{f_{allow}}{F_{tu}} \tag{47}$$

FUSELAGE STRUCTURE FATIGUE

Fuselage cover and minor frame material fatigue life is examined for pressurization cycles and endurance limit. Material endurance limit is used in the fuselage module for the evaluation of acoustic fatigue.

Pressurization Spectrum

Pressure cycle data are user input to this module. As many as 100 maximum-minimum pressure-cycle sets may be used to describe the pressurization during the vehicle life.

Pressure is equated to stress by the hoop stress analogy (equation 48).

$$f = \frac{Pr}{t} \tag{48}$$

where

$P$ = pressure, psi

$r$ = fuselage radius, in.

$t$ = fuselage panel thickness, in.

Similar to the wing approach, the proportionality can be represented by equation 49.

$$f = KP \tag{49}$$

where

$K$ = proportionality constant

The solution for K is identical to that used for the wing. The maximum spectrum pressure is also the static strength design pressure and, therefore, the cutoff stress is calculated by equation 50.

$$f_{allow} = K_2 \, P_{max} \tag{50}$$

where

$K_2$ = proportionality constant which results in spectrum stresses which meet vehicle life requirements

$P_{max}$ = maximum spectrum pressure

This cutoff stress is stored in the material files as a fraction of ultimate tensile strength (equation 45).

Endurance Limit

Endurance limit stress is calculated for the following conditions:

- $K_t$ = 1, polished specimen
- R = -1, fully reversing loading
- n = $10^9$

The initial estimate for $f_{max}$ is half of the ultimate tensile strength. The endurance limit stress is determined by the successive iteration solution used for the wing. The proportionality constant K is also the stress, and life is $10^9$ cycles.

## Section III

## PROGRAM DESCRIPTION

### GENERAL DISCUSSION

The purpose of the fatigue module is to derive the stress level that will give the required life with the specified material so that the factor for ultimate fatigue effect ($KF_{tu}$) can be placed in the material library for use by the weight modules. This calculation can be performed for four distinct conditions.

Two locations on the wing, the side of the fuselage (SOF) and the second wing station (WOS), can be analyzed using a function of bending moments for starting stress levels and cycles based on the corresponding exceedances. These bending moments and exceedances usually come from the loads module, but can be input data.

Two other conditions are analyzed; if the fuselage structure fatigue effect is requested, pressure cycling and endurance limit factors are calculated. Maximum and minimum pressures are used to determine starting stress levels, and the corresponding cycles must also be input. For the endurance limit calculation, one point is set up with maximum and minimum starting stress levels at plus and minus one-half of $F_{tu}$ and for 1 billion cycles. Also, the notch factor and scatter factor are set to 1.0 for the endurance limit, but are determined by the user for the other calculations. If fuselage minor frame fatigue effect is requested, it is computed in the same manner as for the cover.

### PROGRAM FUNCTIONS

The fatigue module consists of a main program (FATGUE), three subroutines, and two function routines. Figure 37 shows the logical flow diagram of this module. The subroutine tree, including system routines READMS and WRITMS, is shown in Figure 38.

### GENERAL MAPS

Data storage and transmittal is accomplished through the use of common, labeled common, and mass storage file records. Mass storage file records are read into, and written from, regions in common. Table 21 presents an alphabetical listing of all program arrays and variables. Table 22 presents descriptions of the input data.

General logic flow:

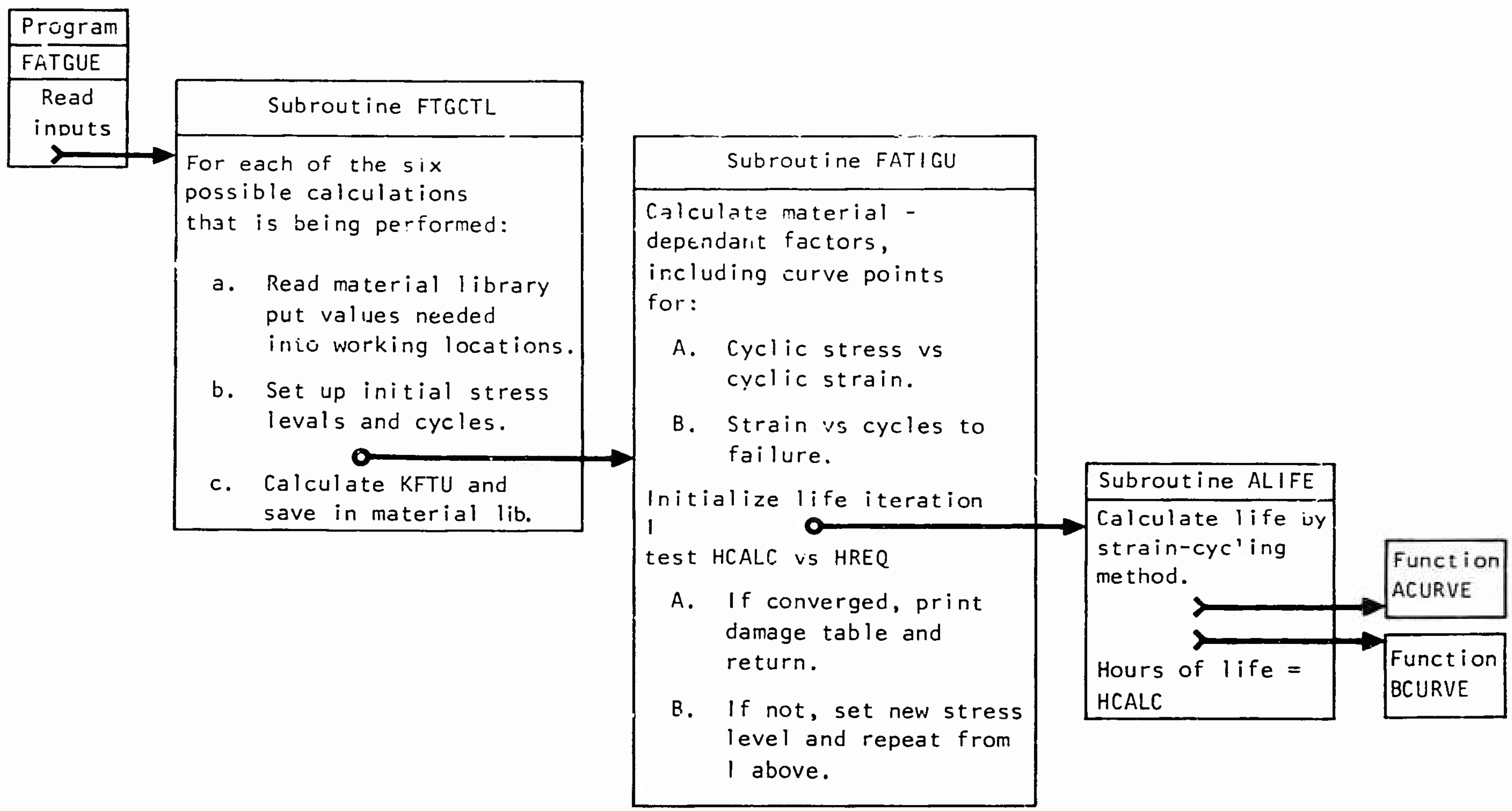



Figure 37. Fatigue module logic flow.

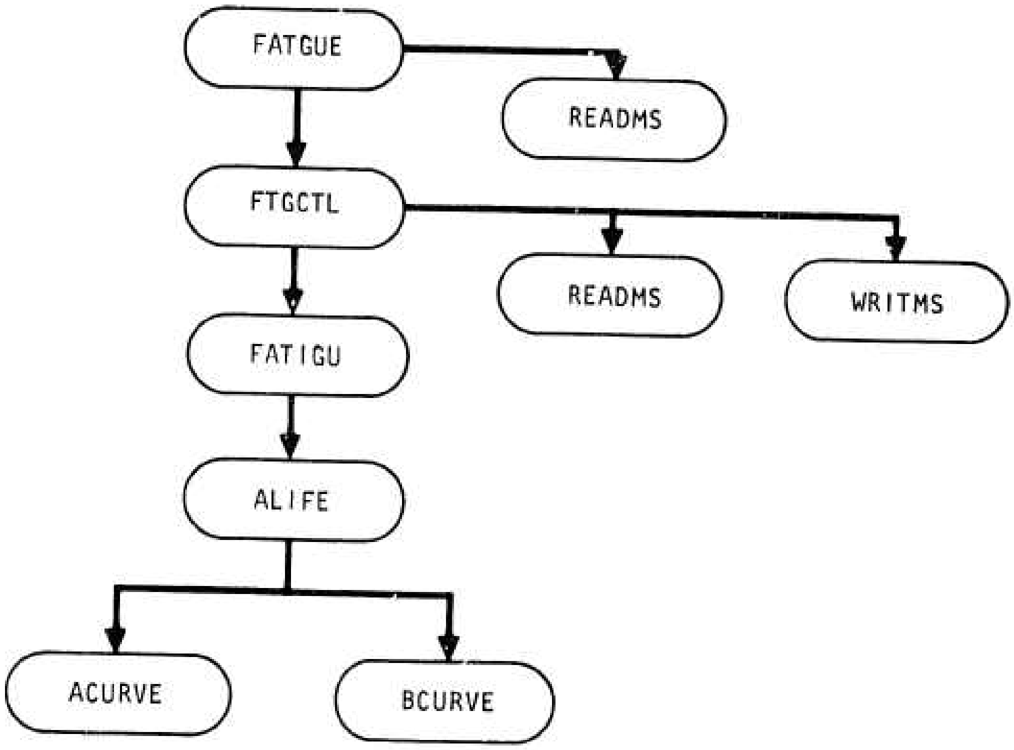



Figure 38. Fatigue module subroutine tree.

TABLE 21. VARIABLE LIST

| Var Name | Size | Common Loc | Type* | Description | Subroutine Reference |
|---|---|---|---|---|---|
| A | 1 | 2451 | C | Upper range estimate of cyclic stress at notch in iteration for SGMX | ALIFE |
| AN | 300 | 801 | C | Applied cycles, set up from exceedances or pressure cycles | FTGCTL, FATIGU |
| ANPT | 1 | 107 | C | Number of points in AN for current condition being calculated | FTGCTL, FATIGU |
| B | 1 | 2452 | C | Lower range estimate of cyclic stress at notch in iteration for SGMX | ALIFE |
| BETA | 1 | 71 | C | Material-dependent factor, β, defining cyclic stress-strain curve | FATIGU, BCURVE |
| BLOCK | 1 | 81 | I | Block size, fraction of required life for each block | FTGCTL, FATIGU |
| BMA | 660 | 1501 | I/IM | Bending moment - gust exceedance array | FATIGUE, FTGCTL |
| BMMX | 1 | 1161 | C | Maximum of FMAX, calculated for wing points only | FTGCTL |
| BMREF | 20 | 2107 | I/IM | 1 g inertia bending moments for each spectra segment for both wing stations | FTGCTL |
| BMSMX | 2 | 1162 | I/IM | Maximum strength design bending moment | FATGUE, FTGCTL |

TABLE 21. VARIABLE LIST (CONT)

| Var Name | Size | Common Loc | Type* | Description | Subroutine Reference |
|---|---|---|---|---|---|
| CON | 50 | 51 | I,C | Input and calculated factors required in strain-cycling fatigue life calculation | FTGCTL, FATIGU, ALIFE, ACURVE, BCURVE |
| CYC | 300 | 2501 | C | Cycles for condition in work, (AN times BLOCK) | FATIGU, ALIFE |
| C1 | 1 | 69 | C | Material-dependent factor, $C_1$, defining cyclic stress-strain curve | FATIGU, BCURVE |
| C1E | 1 | 76 | C | Factor in cyclic stress-strain equation, CON(26) | FATIGU, ALIFE, ACURVE |
| C1X | 10 | 3401 | C | Natural log. of strains for cyclic stress versus cyclic strain equation | FATIGU, ACURVE |
| C1Y | 10 | 3411 | C | Natural log. of stresses for cyclic stress versus cyclic strain equation | FATIGU, ACURVE |
| C2 | 1 | 69 | C | Material dependent factor, $C_2$, defining cyclic stress-strain equation | FATIGU, ALIFE, ACURVE BCURVE |
| C2X | 10 | 3421 | C | Natural log. of strains for strain versus cycles to failure equation | FATIGU, BCURVE |
| C2Y | 10 | 3431 | C | Natural log. of number of cycles to failure for strain versus cycles to failure equation | FATIGU, BCURVE |
| DAM1 | 300 | 3801 | C | Damage ratios for first block | ALIFE, FATIGU |

TABLE 21. VARIABLE LIST (CONT)

| Var Name | Size | Common Loc | Type* | Description | Subroutine Reference |
|---|---|---|---|---|---|
| DAM2 | 300 | 4101 | C | Damage ratios for second block | ALIFE, FATIGU |
| DEPS | 300 | 4401 | C | Strain range for each load cycle in first block | ALIFE |
| DEPSMX | 1 | 2482 | C | Maximum from DEPS array | ALIFE |
| DEPSR2 | 1 | 2483 | C | Incremental residual strain for any load cycle in second block | ALIFE |
| DEPS1 | 1 | 2471 | C | Incremental strain 1 for any cycle in first block | ALIFE |
| DEPS2 | 1 | 2472 | C | Incremental strain 2 for any cycle in first block | ALIFE |
| DF2 | 1 | 2457 | C | Mean nominal positive stress minus minimum nominal stress for a cycle (F1-FMIN) | ALIFE |
| DSGR1 | 1 | 2473 | C | Incremental residual stress 1 for any load cycle in second block | ALIFE |
| DSGR2 | 1 | 2474 | C | Incremental residual stress 2 for any load cycle in second block | ALIFE |
| DSG1 | 1 | 2463 | C | Incremental stress 1 for any cycle in first block | ALIFE |
| DSG2 | 1 | 2464 | C | Incremental stress 2 for any cycle in first block | ALIFE |

TABLE 21. VARIABLE LIST (CONT)

| Var Name | Size | Common Loc | Type* | Description | Subroutine Reference |
|---|---|---|---|---|---|
| DUMMY | 830 | 1501 | I/IM | Block of bending moment and related data from file 35 (refer to Table 4, locations 1501 to 2330) | FATGUE |
| E | 1 | 105 | C | Young's modulus, material-dependent factor | FTGCTL, FATIGU, ALIFE, ACURVE |
| EM | 220 | 2127 | I/IM | Exceedances for maneuver to correspond to bending moments in BMA | FATGUE, FTGCTL |
| EPSA | 1 | 2465 | C | Cyclic strain amplitude for cycle from FMAX to FMIN for any load cycle in first block | ALIFE |
| EPSBT | 1 | 2475 | C | Equivalent strain at edge of notch for fully reversed cycles (R = -1) for any load cycle in second block | ALIFE |
| EPSMX | 1 | 2476 | C | Cyclic maximum strain corresponding to SGMX (at edge of notch) | ALIFE |
| EPSTJ | 1 | 2477 | C | Absolute value of EPST2 | ALIFE |
| EPST2 | 1 | 2478 | C | Equivalent strain at edge of notch for fully reversed cycles (R = -1) for any load cycle in first block | ALIFE |
| F | 1 | 2453 | C | Discrepancy in solution for maximum stress for assumed cyclic stress at notch | ALIFE |

TABLE 21. VARIABLE LIST (CONT)

| Var Name | Size | Common Loc | Type* | Description | Subroutine Reference |
|---|---|---|---|---|---|
| FA | 1 | 2455 | C | Discrepancy in solution for maximum stress for upper range estimate of cyclic stress at notch | ALIFE |
| FACT | 1 | 2497 | C | Multiplying factor for FMAX and FMIN arrays to adjust level for next iteration for life | FATIGU |
| FFIND | 1 | 112 | I/IM | Fuselage minor frame material number indicator:<br>• 0 = set from fuselage module data if it is loaded<br>• + = use value input here<br>• - = do not calculate for fuselage minor frames | FTGCTL |
| FIND | 1 | 111 | I/IM | Fuselage cover material no. indicator:<br>• 0 = set up from fuselage module data<br>• + = use value here<br>• - = do not calculate fuselage cover | FTGCTL |
| FKT | 1 | 103 | C | Working location for elastic stress concentration factor | FTGCTL, ALIFE |
| FKTFP | 1 | 118 | I | Fuselage structure elastic stress concentration factor | FTGCTL |
| FKTW | 2 | 115 | I | Wing elastic stress concentration factor for SOF and WOS | FTGCTL |
| FMAX | 300 | 201 | C | Working locations for maximum nominal stresses | FTGCTL, FATIGU, ALIFE |

TABLE 21. VARIABLE LIST (CONT)

| Var Name | Size | Common Loc | Type* | Description | Subroutine Reference |
|---|---|---|---|---|---|
| FMIN | 300 | 501 | C | Working locations for minimum nominal stresses | FTGCTL, FATIGU, ALIFE |
| FPT | 1 | 1200 | I | Number of points input for fuselage pressures | FTGCTL |
| FSQ | 1 | 2459 | C | Intermediate calculation in iteration for SGMX | ALIFE |
| FTOP | 1 | 2489 | C | Highest initial FMAX used in search to get subscript (NTOP) of that point | FATIGU |
| FTU | 1 | 104 | C | Working location for ultimate tensile strength | FTGCTL, FATIGU |
| F1 | 1 | 2456 | C | Mean nominal positive stress for a cycle, $F_1$ | ALIFE |
| GAMMA | 1 | 72 | C | Material-dependent factor, $\gamma$, defining cyclic stress-strain curve | FATIGU, BCURVE |
| H | 1 | 101 | I/IM | Specified vehicle life | FTGCTL, FATIGU |
| HCALC | 1 | 2500 | C | Calculated hours of vehicle life | ALIFE, FATIGU |
| HN | 1 | 2494 | C | Factor in life iteration, natural log. of HCALC | FATIGU |
| HR | 1 | 2496 | C | Factor in life iteration, natural log. of HREQ | FATIGU |
| HRBL | 1 | 2441 | C | Hours per block | FTGCTL, ALIFE |
| HREQ | 1 | 2499 | C | Required vehicle life, specified life multiplied by scatter factor | FATIGU |

TABLE 21. VARIABLE LIST (CONT)

| Var Name | Size | Common Loc | Type* | Description | Subroutine Reference |
|---|---|---|---|---|---|
| H1 | 1 | 2490 | C | A point in iteration calculation for life, ln (life) | FATIGU |
| H2 | 1 | 2492 | C | Second point in iteration calculation for life, ln (life) | FATIGU |
| I | 1 | -- | C | General index | FATGUE, FTGCTL, ALIFE |
| IFERR | 1 | 5444 | C | Error indicator, nonzero value when failed to find life ND(44) | FTGCTL, FATIGU |
| IFUP | 1 | -- | C | Extrapolation indicator | ACURVE |
| IND | 1 | -- | C | Index for SAVE array | FTGCTL |
| IP | 80 | IPRINT | I | Print/no-print indicators | FATGUE, FTGCTL, FATIGU |
| IREF | 1 | -- | C | Index for 1g bending moments | FTGCTL |
| J | 1 | -- | C | General index | FTGCTL, ALIFE |
| JMAX | 1 | 5441 | C | Index of point with largest incremental strain range (DEPS) to get SGR1MX | ALIFE |
| K | 1 | -- | C | General index | FTGCTL, ACURVE, BCURVE |
| L | 1 | -- | C | Index in bending moments to stresses loop, first point in each spectra | FTGCTL |

TABLE 21. VARIABLE LIST (CONT)

| Var Name | Size | Common Loc | Type* | Description | Subroutine Reference |
|---|---|---|---|---|---|
| LL | 1 | -- | C | Count for input spectra | FTGCTL |
| M | 1 | -- | C | Index in bending moments to stresses loop, last point in each spectra | FTGCTL |
| MI | 1 | 5445 | C | Same or new material indicator, to save repeat calculation | FTGCTL, FATIGU |
| MTIME | 1 | 5446 | C | Counter for number of times try restart if HCALC negative | FATIGU |
| N | 1 | -- | C | General index | FATGUE, FTGCTL, FATIGU, ACURVE, BCURVE |
| ND | 200 | 5401 | C | Fixed indicators | general |
| NMATL | 1 | 5459 | C | Number of material decks in library | FATGUE, FTGCTL |
| NPT | 1 | 5440 | C | Number of points in FMAX FMIN CYC set in work | FTGCTL, FATIGU, ALIFE |
| NSEG | 20 | 5521 | C | Index in FMAX, FMIN, CYC, DAM1, DAM2, etc, to indicate end of each original spectra segment | FTGCTL |
| NTIME | 1 | 5443 | C | Counter for number of iterations for life | FATIGU |
| NTOP | 1 | 5442 | C | Index value for highest FMAX | FTGCTL, FATIGU |
| P | 1 | 75 | C | Intermediate calculation in iteration for SGMX | FATIGU, ALIFE |

TABLE 21. VARIABLE LIST (CONT)

| Var Name | Size | Common Loc | Type* | Description | Subroutine Reference |
|---|---|---|---|---|---|
| PCY | 100 | 1401 | I | Cycles for fuselage pressures | FTGCTL |
| PC1 | 300 | 2801 | C | Percent of total damage for each point in first block | FATIGU |
| PC2 | 300 | 3101 | C | Percent of total damage for each point in second block | FATIGU |
| PMN | 100 | 1301 | I | Minimum pressures for fuselage pressures | FTGCTL |
| PMX | 100 | 1201 | I | Maximum pressures for fuselage pressures | FTGCTL |
| Q | 1 | 53 | C | Material-dependent factor, γ/δ | FATIGU, ALIFE, ACURVE |
| QP1 | 1 | 57 | C | Material-dependent factor, Q+1, power in SGMX calculation | FATIGU, ALIFE |
| RA | 1 | 106 | C | Material property, reduction in area, from library for material in work | FTGCTL, FATIGU |
| RE | 1 | 74 | C | Material-dependent factor, E/1000 | FATIGU |
| RSF | 1 | 2442 | C | Reciprocal of scatter factor | FATIGU, ALIFE |
| RTO | 1 | 65 | C | Material-dependent factor, ductility | FATIGU |
| RTO1 | 1 | 67 | C | Material-dependent factor, FTU/E | FATIGU |

TABLE 21. VARIABLE LIST (CONT)

| Var Name | Size | Common Loc | Type* | Description | Subroutine reference |
|---|---|---|---|---|---|
| RFO2 | 1 | 68 | C | Material dependent factor, 1.0+RFO | FATIGU |
| S | 100 | 2401 | C | Scratch, misc intermediate values | FATIGU, ALIFE |
| SAVE | 16 | -- | C | Print array for old and new values of KFTU to show changes in material record by fatigue calculation | FTGCTL |
| SDAM1 | 1 | 2479 | C | Sum of damages from first block | ALIFE |
| SDAM2 | 1 | 2480 | C | Sum of damages from second block | ALIFE |
| SF | 1 | 102 | C | Scatter factor for condition in work | FTGCTL, FATIGU |
| SFFP | 1 | 117 | I | Scatter factor for fuselage pressure condition | FTGCTL |
| SFM | 1 | 2487 | C | A point in iteration for life, $\ln(\Sigma FMAX)_{prior}$ | FATIGU |
| SFN | 1 | 2495 | C | A point in iteration for life, $\ln(\Sigma FMAX)_{new}$ | FATIGU |
| SFW | 2 | 113 | I | Scatter factors for SOF and WOS | FTGCTL |
| SF1 | 1 | 2491 | C | A point in iteration calculation for life, $\ln(\Sigma FMAX)$ corresponds to H1 | FATIGU |
| SF2 | 1 | 2493 | C | Second point in iteration calculation for life, $\ln(\Sigma FMAX)$ corresponds to H2 | FATIGU |

TABLE 21. VARIABLE LIST (CONT)

| Var Name | Size | Common Loc | Type* | Description | Subroutine Reference |
|---|---|---|---|---|---|
| SGBMEN | 1 | 2484 | C | Cyclic mean stress at notch for a cycle from FMAX to FMIN for second block | ALIFE |
| SGF | 1 | 56 | C | Material-dependent factor, fracture stress | FATIGU, ALIFE |
| SGMEN | 300 | 5001 | C | Cyclic mean stress at notch for a cycle from FMAX to FMIN for first block | ALIFE |
| SGMN | 1 | 2466 | C | Minimum stress at notch for a cycle from FMAX to FMIN | ALIFE |
| SGMX | 1 | 2467 | C | Maximum stress at notch for a cycle from FMAX to FMIN | ALIFE |
| SGR1 | 300 | 4701 | C | Residual stress for each load cycle in first block | ALIFE |
| SGR1MX | 1 | 2485 | C | Point in SGR1 corresponding to DEPSMX; this is residual stress carried over from first to second block | ALIFE |
| SUMF | 1 | 2488 | C | Sum of FMAX values for first pass | FATIGU |
| T | 3600 | 2401 | C | General area for calculated values, scratch | FTGCTL |
| TCOM | 5600 | 1 | C | Common area, cleared in control program | FATGUE |
| TEPSTJ | 1 | 2486 | C | Absolute value of EPST2 and EPSBT for any point | ALIFE |

TABLE 21. VARIABLE LIST (CONT)

| Var Name | Size | Common Loc | Type* | Description | Subroutine Reference |
|---|---|---|---|---|---|
| TF | 1 | 2498 | C | Total multiplying factor used for FMAX, FMIN in life iteration for any condition | FATIGU |
| TMD | 300 | 3501 | IL | Block that material record from library is read into | FRGCTL |
| TNF1 | 1 | 2469 | C | Cycles to failure corresponding to EPST2 | ALIFE |
| TNF2 | 1 | 2470 | C | Cycles to failure corresponding to EPSBT | ALIFE |
| WIND | 2 | 109 | I/IM | Wing material number indicators, (SOF, WOS)<br>• 0 = set up from wing module data if it is loaded<br>• + = use no.'s in input stream<br>• - = do not calculate for this station | FTGCTL |
| X | 1 | 2454 | C | Solution for SGMX, ksi | ALIFE |
| XCL | 1 | 2421 | C | X calculated, ln. | ACURVE |
| XCL | 1 | 2431 | C | X calculated, ln. | BCURVE |
| XCR | 1 | 2428 | C | X calculated, real | ACURVE |
| XCR | 1 | 2438 | C | X-calculated, real | BCURVE |
| XI | 1 | 2420 | C | Curve point used in interpolation ln of input x | ACURVE |

TABLE 21. VARIABLE LIST (CONT)

| Var Name | Size | Common Loc | Type* | Description | Subroutine Reference |
|---|---|---|---|---|---|
| XI | 1 | 2430 | C | Curve point used in interpolation In of input x | BCURVE |
| XJ | 1 | 2422 | C | Curve point used in interpolation | ACURVE |
| XJ | 1 | 2432 | C | Curve point used in interpolation | BCURVE |
| XK | 1 | 2424 | C | Curve point used in interpolation | ACURVE |
| XK | 1 | 2434 | C | Curve point used in interpolation | BCURVE |
| XMISC | 100 | MISC | LM | Labeled common array used to convey single items from one module to another | FATGUE, FTGCTL |
| XPRIOR | 1 | -- | C | Value of previous x-argument | ACURVE |
| YI | 1 | 2426 | C | Curve point calculated by interpolation correspond to XI | ACURVE |

TABLE 21. VARIABLE LIST (CONCL)

| Var Name | Size | Common Loc | Type* | Description | Subroutine Reference |
|---|---|---|---|---|---|
| YI | 1 | 2436 | C | Curve point calculated by interpolation, correspond to XI | BCURVE |
| YJ | 1 | 2423 | C | Curve point used in interpolation | ACURVE |
| YJ | 1 | 2433 | C | Curve point used in interpolation | BCURVE |
| YK | 1 | 2425 | C | Curve point used in interpolation | ACURVE |
| YK | 1 | 2435 | C | Curve point used in interpolation | BCURVE |
| YPRIOR | 1 | -- | C | Values of y-calculated for XPRIOR | ACURVE |
| YREAL | 1 | 2427 | C | Real value of interpolated value to check in stress-strain equation | ACURVE |
| YREAL | 1 | 2437 | C | Real value of interpolated value to check in strain versus cycles equation | BCURVE |

*Symbols used to designate type represent:

C - stored during calculation
I - input through data stream
IL - input from library files
IM - input from another module

TABLE 22. FATIGUE INPUT DATA

| D Loc | Var Name | Description ((I/L) = Input or From Loads Module) |
|---|---|---|
| 1 | | 1.0 |
| 2 | | 2.0 |
| 3 | | 3.0 |
| 4 | | 4.0 |
| 5 | | 5.0 |
| 6 | | 6.0 |
| 7 | | 7.0 |
| 8 | | 8.0 |
| 9 | | 9.0 |
| 10 | | 10.0 |
| 11 | | 11.0 |
| 12 | | 12.0 |
| 13 | | 20.0 |
| 14 | | 1000.0 |
| 15 | | 3.1415927, pi |
| 16 | | 0.01745329, deg to radians factor |
| 17 | | 144.0 |
| 18 | | 24.0 |
| 19 | | 0.5 |
| 20 | | 1.5 |
| 21 | | 0.3333333 |
| 22 | | 0.95 |
| 23 | | 0.25 |
| 24 | | 0.0 |
| 25 through 50 | | Not used |

TABLE 22. FATIGUE INPUT DATA (CONT)

| D Loc | Var Name | Description ((I/L) = Input or From Loads Module) |
|---|---|---|
| 51 | CON(1) | 1.12 |
| 52 | CON(2) | 0.893 |
| 53 | CON(3) | 0.0792 |
| 54 | CON(4) | 0.179 |
| 55 | CON(5) | 3.31 |
| 56 | CON(6) | 0.25 |
| 57 | CON(7) | 81.4 |
| 58 | CON(8) | 0.75 |
| 59 | CON(9) | $1.0 \times 10^8$ |
| 60 | CON(10) | $1.0 \times 10^6$ |
| 61 | CON(11) | 0.015, tolerance in $\sigma_{MAX}$ root iteration - ALIFE |
| 62 | CON(12) | 0.001, tolerance in ACURVE and BCURVE |
| 63 | CON(13) | 0.01, tolerance in iteration for life - FATIGU |
| 64 | CON(14) | -0.1666667, factor in '2nd stress' level estimate - FATIGU in life iteration |
| 65 through 77 | | Used for calculated values of material-dependent data |
| 78 | CON(28) | 0.5, factor in initial stress level - FTGCIL |
| 79 | CON(29) | $1.0 \times 10^9$, cycles for endurance limit |
| 80 | CON(30) | 1.0, notch factor for endurance limit |
| 81 | BLOCK | 0.05, block size, fraction of required life |
| 82 through 100 | | Not used |
| 101 | H | Hours of vehicle life (I or trans from Data Management) |
| 102 through 107 | | Used for calculations |

TABLE 22. FATIGUE INPUT DATA (CONT)

| D Loc | Var Name | Description ((I/L) - Input or From Loads Module) |
|---|---|---|
| 108 | | Not used |
| 109 | WIND(1) | SOF, wing |
| 110 | WIND(2) | WOS, wing |
| 111 | FIND | Fuselage cover |
| 112 | FFIND | Fuselage minor frames |
| | | Material number indicator (108–112):<br>• 0 value = set from corresponding weight module data<br>• + value = used from FATIGUE input<br>• - value = no fatigue calculation for that item |
| 113 | SFW(1) | Wing scatter factor for SOF |
| 114 | SFW(2) | Wing scatter factor for WOS |
| 115 | FKTW(1) | Wing notch factor for SOF |
| 116 | FKTW(2) | Wing notch factor for WOS |
| 117 | SFFP | Scatter factor for fuselage pressure cycles |
| 118 | FKTFP | Notch factor for fuselage pressure cycles |
| 119 through 200 | | Not used |
| 201 through 1100 | | Set up during calculation |
| 1101 through 1160 | | Not used |
| 1161 | | Used for calculated variable BMMX |
| 1162 | BMSMX(1) | (I/L) maximum static bending moment SOF |
| 1163 | BMSMX(2) | (I/L) maximum static bending moment WOS |
| 1164 through 1199 | | Not used |
| 1200 | FPT | Number of points in fuselage pressure array (1 to 100) |
| 1201 through 1300 | FPR(1) through FPR(100) | Maximum fuselage pressures, PMX |

TABLE 22. FATIGUE INPUT DATA (CONT)

| D Loc | Var Name | Description ((I/L) - Input or From Loads Module) |
|---|---|---|
| 1301 through 1400 | FPR(101) through FPR(200) | Minimum fuselage pressures, PMN, corresponding to PMX |
| 1401 through 1500 | FPR(201) through FPR(300) | Applied cycles for fuselage pressures, PCY, corresponding to PMX, PMN |
| 1501 | BMA(1,1) | (I/L) Maximum bending moment, SOF, 1st point, 1st spectra seg |
| 1502 | BMA(2,1) | (I/L) Maximum bending moment, WOS, 1st point, 1st spectra seg |
| 1503 | BMA(3,1) | (I/L) Exceedance for gust, 1st point, 1st spectra seg |
| 1504 | BMA(1,2) | (I/L) Maximum bending moment, SOF, 2nd point, 1st spectra seg |
| through | through | |
| 1558 | BMA(1,20) | (I/L) Minimum bending moment, SOF, 20th point, 1st spectra seg |
| 1559 | BMA(2,20) | (I/L) Minimum bending moment, WOS, 20th point, 1st spectra seg |
| 1560 | BMA(3,20) | (I/L) Exceedance for gust, 20th point, 1st spectra seg |
| 1561 | BMA(1,21) | (I/L) Maximum bending moment, SOF, 1st point, 2nd spectra seg |
| through 2100 | through BMA(3,200) | (I/L) Exceedance for gust, 20th point, 10th spectra seg |
| 2101 | BMA(1,201) | (I/L) Positive bending moment, SOF, Ground-air-ground spectra seg |
| 2102 | BMA(2,201) | (I/L) Positive bending moment, WOS, Ground-air-ground spectra seg |
| 2103 | BMA(3,201) | (I/L) Occurrences, Ground-air-ground spectra seg |
| 2104 | BMA(1,202) | (I/L) Negative bending moment, SOF, Ground-air-ground spectra seg |
| 2105 | BMA(2,202) | (I/L) Negative bending moment, WOS, Ground-air-ground spectra seg |

TABLE 22. FATIGUE INPUT DATA (CONCL)

| D Loc | Var Name | Description ((I/L) - Input or From Loads Module) |
|---|---|---|
| 2106 | BMA(3,202) | (I/L) Occurrences, ground-air-ground spectra seg |
| 2107 through 2116 | BMREF(1) through BMREF(10) | (I/L) 1 g net bending moments, SOF, for spectra segments 1 through 10 |
| 2117 through 2126 | BMREF(11) through BMREF(20) | (I/L) 1 g net bending moments, WOS, for spectra segments 1 through 10 |
| 2127 through | EM(1) through | (I/L) Exceedances for maneuver, 20 for each spectra segment, only 1 through 10 have data as not used on ground-air-ground |
| 2346 | EM(220) | (after 202 could be used) |
| 2347 through 2400 | | Not used |

COMMON

Common consists of 5,600 cells which are divided into the major regions, as follows:

| Common Location | Variable Name | Description |
|---|---|---|
| 1 through 2400 | D(1) D(2400) | Physical constants, equation constants, input data, and material-dependent factors |
| 2401 through 5400 | T(1) T(3000) | Calculated variables and storage of material library file data |
| 5401 through 5600 | ND(1) ND(200) | Storage regio[illegible] indicators and counters |

## LABELED COMMON

Labeled common arrays are used to transfer program control words and certain vehicle design data. These arrays are as follows:

1. XMISC (Block MISC) - This array is used to transmit certain vehicle design data as shown in Table 23.

2. IP (Block IPRINT) - This array is used to transmit the following print controls:

   a. IP(56) - This indicator is used in subroutine FATGUE to designate output print of bending moment exceedance data. (See Figure 39.)

   b. IP(57) - This indicator is used in subroutines FTGCTL and FATIGU to designate output print of input factors, initial stress-occurrence array, and final damage tables. (See Figures 40 through 44.)

   c. IP(58) - This indicator is used in subroutine FATIGU to designate output print of intermediate program calculations. (See Figures 45 and 46.)

   For Figure 45, the data are arranged in the following manner:

   1. Three lines of material-dependent factors from CON array:

      | | | | | | |
      |---|---|---|---|---|---|
      | 65 | RTO | SEG | RT01 | RT02 | C1 |
      | 70 | C2 | BETA | GAMMA | Q | RE |
      | 75 | P | C1E | QPI | - | - |

   2. Four columns of calculated coordinates for stress-strain curve and strain versus cycles to failure curve, and natural logarithms of the coordinates as used for interpolation:
      C1X C1Y C2X C2Y

   For Figure 46, the three lines of printed data are arranged in the following manner:

   NTIME, HCALC, FACT, TF
   H1, SF1, H2, SF2
   HN, SFN, HR, *

   *Natural logarithm of FACT for NTIME = 1 only.

TABLE 23. XMISC ARRAY VARIABLES (MISC BLOCK)

| Loc | Description | Subroutine Reference |
|---|---|---|
| 1 | Number of material records in material library | FATGUE |
| 2 through 14 | Controls and design data used by other program modules | |
| 15 | Wing structure material identification number; established in executive module | FTGCTL |
| 16 through 30 | Controls and design data used by other program modules | |
| 31 | Fuselage cover material identification number; established in executive module | FTGCTL |
| 32 | Maximum net unswept bending moment at wing side of fuselage station (in.-lb); established in airload module | FATGUE |
| 33 | Maximum net swept bending moment at outboard wing fatigue evaluation station (in.-lb); established in airload module | FATGUE |
| 34 | Required vehicle life (hr); established in data management module | FATGUE |
| 35 through 40 | Controls and design data used by other program modules | |
| 41 | Fuselage minor frame material identification number; established in executive module | FTGCTL |
| 42 through 100 | Controls and design data used by other program modules | |

*** BC(3,220) FROM SUBROUTINE FATMG IN LOADS PROGRAM IN FLCORD 35 ***    ** FATGUE - IP(56) **

| | SOF BEND MOM | WOS BEND MOM | EXCEEDANCES-GUST | EXCEEDANCES-MANU |
|---|---|---|---|---|
| 1 | 238564720. | 252834312. | 0.72702E-42 | 0.0 |
| 2 | 132467120. | 232674256. | 0.50260E-37 | 0.0 |
| 3 | 175560320. | 212413600. | 0.34746E-32 | 0.75000E-01 |
| 4 | 153472736. | 192652032. | 0.24020E-27 | 0.42500E+00 |
| 5 | 142975600. | 172932320. | 0.16605E-22 | 0.22500E+01 |
| 6 | 126478400. | 153031664. | 0.11480E-17 | 0.10000E+02 |
| 7 | 109581216. | 133071108. | 0.79363E-13 | 0.47500E+02 |
| 8 | 93484016. | 113110336. | 0.54866E-08 | 0.26250E+03 |
| 9 | 76586816. | 93149696. | 0.37330E-03 | 0.36750E+04 |
| 10 | 60489630. | 73188072. | 0.26765E+02 | 0.17500E+06 |
| 11 | 43491536. | 53831952. | 0.26765E+02 | 0.20000E+05 |
| 12 | 32594352. | 33271296. | 0.37930E-03 | 0.34000E+03 |
| 13 | 16497183. | 19960640. | 0.54866E-08 | 0.75000E+00 |
| 14 | 0. | 0. | 0.79363E-13 | 0.25000E-02 |
| 15 | -16497183. | -19960640. | 0.11480E-17 | 0.0 |
| 16 | -32594352. | -33271296. | 0.16606E-22 | 0.0 |
| 196 | -10837900. | -8410162. | 0.22500E+02 | 0.0 |
| 197 | -8128425. | -6307622. | 0.75000E+00 | 0.0 |
| 198 | -5418950. | -4205081. | 0.25000E-01 | 0.0 |
| 199 | -4064212. | -3153810. | 0.50000E-02 | 0.0 |
| 200 | -2709475. | -2102541. | 0.10000E-02 | 0.0 |
| 201 | 68738256. | 83168376. | 0.25000E+05 | 0.0 |
| 202 | -64562304. | -50146560. | 0.25000E+05 | 0.0 |

REFERENCE BENDING MOMENTS FOR MANEUVER

| SEGMENT | SOF | WOS |
|---|---|---|
| 1 | 54990608. | 66535504. |
| 2 | 46880208. | 59430400. |
| 3 | 50366784. | 68794464. |
| 4 | 43902768. | 64913616. |
| 5 | 51637872. | 68770848. |
| 6 | 45424368. | 59545680. |
| 7 | 45904160. | 59534400. |
| 8 | 46685024. | 59473472. |

RMSMX(1) = 320510464.
RMSMX(2) = 253423024.

Figure 39. Sample output - bending moment - exceedance data.

FATIGUE INPUT DATA ** FTGCTL - IP(57) **

F= 0.5000000E+05 SF= 0.4000000E+01 KT= 0.3000000E+01

FTU= 0.1290000E+06 E= 0.1640022E+08 PA= 0.2700000E+00 NPT= 157

WING OUTBOARD STATION

STRESS LEVELS SET UP FROM BENDING MOMENTS TIMES 0.260627E-03

| | | | |
|---|---|---|---|
| 1 | 0.6721781E+05 | -0.2987461E+05 | 0.5025913E-37 |
| 2 | 0.6161629E+05 | -0.2427312E+05 | 0.3474500E-32 |
| 3 | 0.6161628E+05 | 0.1867163E+05 | 0.7499999E-01 |
| 4 | 0.5601488E+05 | -0.1867162E+05 | 0.2401981E-27 |
| 5 | 0.5601488E+05 | 0.1867163E+05 | 0.3500000E+00 |
| 6 | 0.5041340E+05 | -0.1307014E+05 | 0.1660504E-22 |
| 7 | 0.5041340E+05 | 0.1867163E+05 | 0.1824999E+01 |
| 8 | 0.4481191E+05 | -0.7468648E+04 | 0.1179525E-17 |
| 9 | 0.4481191E+05 | 0.1867163E+05 | 0.7750001E+01 |
| 10 | 0.3921043E+05 | -0.1867162E+04 | 0.7936224E-13 |
| 145 | 0.3514366E+05 | -0.1668983E+04 | 0.8785712E-08 |
| 146 | 0.3514366E+05 | 0.1668984E+05 | 0.2375000E+01 |
| 147 | 0.1668984E+05 | -0.1668983E+04 | 0.2499999E-02 |
| 148 | 0.3004170E+05 | 0.3337966E+04 | 0.5370368E-04 |
| 149 | 0.3004170E+05 | 0.1668984E+05 | 0.6000000E+02 |
| 150 | 0.1668984E+05 | 0.3337966E+04 | 0.3724997E+00 |
| 151 | 0.2503475E+05 | 0.8344914E+04 | 0.3288534E+00 |
| 152 | 0.2503475E+05 | 0.1668984E+05 | 0.2000000E+04 |
| 153 | 0.1668984E+05 | 0.8344914E+04 | 0.5462500E+02 |
| 154 | 0.2002780E+05 | 0.1335186E+05 | 0.2292679E+05 |
| 155 | 0.2002780E+05 | 0.1668984E+05 | 0.7293750E+05 |
| 156 | 0.1668984E+05 | 0.1335186E+05 | 0.4745000E+04 |
| 157 | 0.2333954E+05 | -0.1407246E+05 | 0.2500000E+05 |

Figure 40. Sample output - fatigue input data.

MASS STORAGE FILES

Mass storage file records used in this program are described in Table 24.

TABLE 24. MASS STORAGE FILE RECORDS

| Record No. | Variable and Length | Write Routine | Read Routine | Description |
|---|---|---|---|---|
| 29 | D(2400) | Executive module | FATGUE | Fatigue module input data file |
| 35 | DUMMY (830) | Executive and airload modules | FATGUE | Wing bending moment fatigue spectra data |
| 41 through 60 | TMD(500) | Executive module | FTGCTL | Material library files for 20 different materials |

SUBROUTINE DESCRIPTIONS

PROGRAM FATGUE

General Description

Deck name: FATGUE
Entry name: OVERLAY (5HALPHA,5,0)
Called by: PROGRAM OLAY00
Subroutines called: FTGCTL

This is the control program for the fatigue module. This routine reads the required data and calls FTGCTL to perform the calculations.

The data are read from record 29. The maximum static bending moments for the wing at the side of the fuselage (SOF) and the wing outer panel station (WOS) are initialized from data if present or from values saved during the loads calculation. Bending moments and exceedances calculated in the loads module or input with fatigue data are read from record 35, and are printed if IP(56) is zero.

NUMBER OF ITERATIONS IN SUBROUTINE FATIGU = 3 ** FATIGU - IP(57) **

CALC. LIFE(HR) REQUIRED LIFE(HR)
0.103737E+16 0.200000E+06 HIGHEST FMAX= 0.605889E+05 = 43.59PCT FTU

| N | FMAX | FMIN | APP.CYC. | DAMAGE 1 | PCT | DAMAGE 2 | PCT |
|---|---|---|---|---|---|---|---|
| 1 | 59592. | -26044. | 0. | 0.420143E-41 | 0.00 | 0.427119E-41 | 0.00 |
| 2 | 53716. | -21161. | 0. | 0.175451E-36 | 0.00 | 0.221442E-36 | 0.00 |
| 3 | 53716. | 16278. | 0. | 0.344537E-06 | 0.00 | 0.613378E-05 | 0.19 |
| 4 | 49833. | -16278. | 0. | 0.648662E-32 | 0.00 | 0.106809E-31 | 0.00 |
| 5 | 49833. | 16278. | 0. | 0.670641E-06 | 0.00 | 0.121253E-04 | 0.38 |
| 6 | 43350. | -11394. | 0. | 0.197909E-27 | 0.00 | 0.446992E-27 | 0.00 |
| 7 | 43350. | 16278. | 0. | 0.107857E-05 | 0.00 | 0.202511E-04 | 0.64 |
| 8 | 39066. | -6511. | 0. | 0.447830E-23 | 0.00 | 0.142544E-22 | 0.00 |
| 9 | 39066. | 16278. | 0. | 0.380615E-06 | 0.00 | 0.158449E-04 | 0.50 |
| 10 | 34183. | -1628. | 0. | 0.610379E-19 | 0.00 | 0.260038E-18 | 0.00 |
| 11 | 34183. | 16278. | 2. | 0.405906E-06 | 0.00 | 0.525043E-05 | 0.17 |
| 12 | 16278. | -1628. | 0. | 0.991924E-12 | 0.00 | 0.200420E-11 | 0.00 |
| 13 | 22300. | 3256. | 0. | 0.354150E-15 | 0.00 | 0.148309E-14 | 0.00 |
| 14 | 22300. | 16278. | 11. | 0.801767E-07 | 0.00 | 0.551655E-06 | 0.02 |
| 149 | 26130. | 14550. | 3. | 0.466546E-08 | 0.00 | 0.207360E-07 | 0.00 |
| 150 | 14550. | 2910. | 0. | 0.267166E-11 | 0.00 | 0.540217E-11 | 0.00 |
| 151 | 21825. | 7275. | 0. | 0.684729E-10 | 0.00 | 0.185777E-09 | 0.00 |
| 152 | 21825. | 14550. | 100. | 0.958663E-09 | 0.00 | 0.287346E-08 | 0.00 |
| 153 | 14550. | 7275. | 3. | 0.559942E-11 | 0.00 | 0.117725E-10 | 0.00 |
| 154 | 17460. | 11640. | 1146. | 0.542415E-09 | 0.00 | 0.127038E-08 | 0.00 |
| 155 | 17460. | 14550. | 3647. | 0.239417E-11 | 0.00 | 0.577302E-11 | 0.00 |
| 156 | 14550. | 11640. | 237. | 0.831622E-13 | 0.00 | 0.181164E-12 | 0.00 |
| 157 | 23347. | -12268. | 1250. | 0.195677E-02 | 0.78 | 0.305322E-02 | 96.35 |
| CUMULATIVE DAMAGE FOR EACH BLOCK | | | | 0.196257E-02 | 0.79 | 0.314390E-02 | 99.21 |

NUMBER OF RESIDUAL DAMAGE BLOCKS 78.895

Figure 41. Sample output - final damage table for wing outboard station.

N FOR END OF EACH SPECTRA SEGMENT FOLLOWS ** FTGCTL - IP(57) **

| | |
|---|---|
| 1 | 21 |
| 2 | 42 |
| 3 | 61 |
| 4 | 90 |
| 5 | 99 |
| 6 | 119 |
| 7 | 137 |
| 8 | 156 |
| 9 | 156 |
| 10 | 156 |

NOTE Each of the spectra segments in Figure 10 is defined by 20 bending moment exceedance points. This table correlates the number of pertinent bending moment occurrance sets obtained with that spectra data. N given here is the index shown in Figures 11 and 12.

Figure 42. Sample output - index at the end of each spectra segment.

```
FATIGUE   INPUT DATA                                                    ** FTGCTL - IP(57) **

H= 0.5100000E+05       SF= 1.4000000E+01      KT= 0.3000000E+01

FTU= 0.1390000E+06      F= 0.1643022E+08      RA= 0.2700000E+00                 NPT=    1


FUSELAGE COVER

STRESS LEVELS SET UP FROM FUSELAGE PRESSURES TIMES  0.315909E+04


      1        0.3474995E+05       0.0                    0.2500000E+05


NUMBER OF ITERATIONS IN SUBROUTINE FATIGU =    3                        ** FATIGU - IP(57) **
      CALC. LIFE(HR)             REQUIRED LIFE(HR)
      0.199504E+06                0.200000E+06      HIGHEST FMAX= 0.290857E+05 =  20.92PCT FTU

   N      FMAX        FMIN     APP.CYC.     DAMAGE 1      PCT     DAMAGE 2      PCT
   1     29086.        0.        1250.     0.313277E-02   1.25   0.313277E-02   98.75

      CUMULATIVE DAMAGE FOR EACH BLOCK     0.313277E-02   1.25   0.313277E-02   98.75
NUMBER OF RESIDUAL DAMAGE BLOCKS    78.801
```

Figure 43. Sample output - fuselage pressure cycle input and damage table.

```
ENDURANCE LIMIT
     1        0.6650000E+05      -0.6650000E+05      0.1000000E+10


NUMBER OF ITERATIONS IN SUBROUTINE FATIGU =    3                         ** FATIGU - IP(57) **

     CALC. LIFE(HR)            REQUIRED LIFE(HR)
      0.4599952E+05             0.5000000E+05      HIGHEST FMAX= 0.247820E+05 =  17.83PCT FTU

  N        FMAX        FMIN      APP.CYC.        DAMAGE 1      PCT        DAMAGE 2       PCT

  1       24782.     -24782.    50000000.     0.500007E-01    5.00     0.500007E-01    95.00

       CUMULATIVE DAMAGE FOR EACH BLOCK       0.500007E-01    5.00     0.500007E-01    95.00
NUMBER OF RESIDUAL DAMAGE BLOCKS    19.000
```

Figure 44. Sample output - endurance limit starting values and damage table.

| | | | | | |
|---|---|---|---|---|---|
| 65 | 0.3147104E+00 | 0.1827440E+06 | 0.8475497E-02 | 0.1314710E+01 | 0.1211987E-01 |
| 70 | 0.2001215E+00 | 0.1004705E+00 | 0.5809507E+03 | 0.5782299E+01 | 0.1640021E+05 |
| 75 | 0.1684268E-09 | 0.1987686E+06 | 0.6787239E+01 | 0.5000000E+00 | 0.1000000E+10 |

CURVE SET-UP

| | | | |
|---|---|---|---|
| 0.6057480E-07 | 0.1000000E+01 | 0.2122413E+00 | 0.1000000E+01 |
| 0.2322958E-05 | 0.3809702E+02 | 0.6213898E-01 | 0.1000000E+02 |
| 0.8849776E-04 | 0.1451383E+04 | 0.2141511E-01 | 0.1000000E+03 |
| 0.3434025E-02 | 0.5529337E+05 | 0.9672381E-02 | 0.1000000E+04 |
| 0.1348615E-01 | 0.1105867E+06 | 0.5753633E-02 | 0.1000000E+05 |
| 0.8043385E-01 | 0.1658801E+06 | 0.4061129E-02 | 0.1000000E+06 |
| 0.1605555E+05 | 0.1401294E+07 | 0.2417130E-02 | 0.1000000E+08 |
| 0.3667668E+10 | 0.1183762E+08 | 0.1199237E-02 | 0.9999999E+10 |
| 0.8376300E+15 | 0.1000000E+09 | 0.4752390E-03 | 0.9999998E+14 |
| | | 0.1494713E-03 | 0.9999997E+19 |

| | | | |
|---|---|---|---|
| -16.612793 | 0.0 | -1.550031 | 0.0 |
| -12.972670 | 3.640136 | -2.778381 | 2.302585 |
| -9.332534 | 7.280272 | -3.843658 | 4.605170 |
| -5.656701 | 10.920409 | -4.638481 | 6.907756 |
| -4.306092 | 11.613556 | -5.157924 | 9.210340 |
| -2.520320 | 12.015021 | -5.506294 | 11.512926 |
| 9.684059 | 14.152907 | -6.025174 | 16.118088 |
| 22.022812 | 16.286789 | -6.726069 | 23.025849 |
| 34.361588 | 18.420670 | -7.551693 | 32.236191 |
| | | -8.908406 | 43.749115 |

Figure 45. Sample output - material-dependent factors and curve coordinate.

```
                                                                      ** FATIGU - IP(58) **

***** NTIME=  1        HCALC=  0.4948071E+05        FACTOR=  0.7923194E+00  TF=  0.7923194E+00
      0.1080933E+02     0.1552383E+02     0.0                 0.0
      0.0               0.1529704E+02     0.1220607E+02      -0.2327908E+00

***** NTIME=  2        HCALC=  0.5758477E+06        FACTOR=  0.1105511E+01  TF=  0.8759175E+00
      0.1080933E+02     0.1552383E+02     0.1326360E+02       0.1529704E+02
      0.1326360E+02     0.1539735E+02     0.1220607E+02      -0.3237671E+05

***** NTIME=  3        HCALC=  0.1838914E+06        FACTOR=  0.9952809E+00  TF=  0.8717839E+00
      0.1215373E+02     0.1533735E+02     0.1326360E+02       0.1529704E+02
      0.1215373E+02     0.1533262E+02     0.1220607E+02      -0.3537054E+05
```

Figure 46. Sample output - intermediate values in life interpolation calculation.

## Arrays and Variables Used

BMA, BMREF, BMSMX, D, DUMMY, EM, IP, NMATL, TCOM, XMISC

## Arrays and Variables Calculated

None

## Scratch Arrays

None

## Labeled Common Arrays

IP, XMISC

IP(56) - controls printing of bending moments and exceedances, reference bending moments, and maximum static bending moments.

XMISC(1) - the number of material records in the material library, is placed in NMATL.

XMISC(32) - the maximum static bending moment for SOF from the loads module, is placed in BMSMX(1) if not input in data stream.

XMISC(33) - the maximum static bending moment for WOS from the loads module, is placed in BMSMX(2) if not input in data stream.

XMISC(34) - required vehicle life in hours (H) is put in D(101).

## Mass Storage File Records Used

Record 29 - data entered through the input stream (Table 22).

Record 35 - bending moments, exceedances, etc, usually from the loads module routine FATMG, locations 1501 through 2346 in Table 22.

Error Messages

None

SUBROUTINE FTGCTL

General Description

Deck name: FTGCTL
Entry name: FTGCTL
Called by: FATGUE
Subroutine called: FATIGU

This routine performs the setup operations for the four conditions that can be handled, and gives a summary of the $KF_{tu}$ values calculated. (See Figure 47.)

For the two wing stations, initial stress levels are calculated from the bending moments and exceedances which usually come from the loads module. Each spectra segment has 20 points; each bending moment has a gust exceedance and a maneuver exceedance. Preliminary stress values and the number of occurrences are obtained from these in the following manner:

Let

B = bending moments

gB = 1 g bending moment for this spectra segment

G = gust exceedances

M = maneuver exceedances

F' = preliminary stresses

C = occurrences

then

$$F'_{max_1} = B_2 + (B_1 - B_2)/3$$

$$F'_{min_J} = B_{19} + (B_{20} - B_{19})/3$$

```
CHANGES MADE TO MATERIAL PROPERTIES BY FATIGUE PROGRAM                              ** FTGCTL **

           SIDE OF FUSELAGE ** MATL NO  0.
       TMP(133) CHANGED FROM  0.0     TO    0.0

            WING STATION 2 ** MATL NO 13.
       TMP(134) CHANGED FROM  1.0000 TO    0.4460

            FUSELAGE COVER ** MATL NO 13.
       TMP(131) CHANGED FROM  0.2800 TO    0.1783
       TMP(132) CHANGED FROM  0.5000 TO    0.2092

         FUSELAGE MINOR FRAME ** MATL NO 13.
       TMP(131) CHANGED FROM  0.2800 TO    0.1783
       TMP(132) CHANGED FROM  0.5000 TO    0.2092
```

Figure 47. Sample output - summary of $KF_{tu}$ values put into the library.

$$C_1 = G_2 - G_1 \qquad \text{due to symmetry}$$

$$\left.\begin{aligned} F'_{max_2} &= B_2 + (B_1 - B_2)/3 \\ F'_{min_2} &= gb \end{aligned}\right\} \text{or max and min switched, depending on which is larger}$$

$$C_2 = M_2 - M_1$$

$$\left.\begin{aligned} F'_{max_3} &= gb \\ F'_{min_3} &= B_{19} + (B_{20} - B_{19})/3 \end{aligned}\right\} \text{or max and min switched, depending on which is larger}$$

$$C_3 = M_{19} - M_{20}$$

If $F'_{max_1}$ is zero or negative, that point is not used and the next set of B-points is tried. Also, if any C-value is zero or negative, the corresponding $F'$ values are not used. The $F'_{max}$ and $F'_{min}$ arrays are always kept compact, and a total count is thus generated.

The initial stress level to use for the life iteration is:

$$FMAX_i = F'_{max_i} \left(1/2 \; FTU/F''_{max}\right)$$

and

$$FMIN_i = F'_{min_i} \left(1/2 \; FTU/F''_{max}\right)$$

where

$F''_{max}$ is the maximum of all $F'_{max}$ values

For fuselage pressure cycling, the initial stress level is calculated from the input values:

$PMX_i$ = maximum pressure

$PMN_i$ = minimum pressure

$PCY_i$ = occurrences of this max-min cycle

then

$$FMAX_i = PMX_i \; (1/4 \; FTU/PMX_{max})$$

and

$$FMIN_i = PMN_i \; (1/4 \; FTU/PMX_{max})$$

$$C_i = PCY_i$$

where $PMX_{max}$ is the maximum of all PMX values.

$KF_{tu}$ ultimate fatigue factor is calculated by this routine after the life calculation has reached convergence.

For the wing stations

$$KF_{tu} = (f_{max}/F_{tu})(BM/B_{max})$$

where

$f_{max}$ = highest value in the FMAX array after adjustment to give the desired life

BM = maximum static bending moment for that station

$B_{max}$ = maximum value in the preliminary stress array ($F''_{max}$ in the foregoing discussion)

$F_{tu}$ = ultimate tensile strength

For the fuselage cover, factor for pressure cycles and endurance limit is:

$$KF_{tu} = f_{max}/F_{tu}$$

Logic Flow Outline

1. Calculate for SOF(WOS) material record is read (if necessary) and FTU, RA, and I are placed in working locations.

   a. Set up SAVE with KFTU value and material number from library.

   b. Calculate initial stress and occurrences.

   c. Put scatter factor, notch factor, and number of stress points in working locations.

   d. Print factors and stress-occurrences array if IP(57) = 0.

   e. Call FATIGH.

   f. Calculate ultimate fatigue factor, and place in SAVE and in material working area

   g. Put material record back in library.

2. Repeat from step 1 for WOS.

3. Calculate for fuselage cover, read material (if necessary), and put FTU, RA, and I in working locations.

   a. Put library value of KFTU and material number in SAVE.

   b. Put scatter factor, notch factor, and number of stress points in working locations.

   c. Calculate initial stress and occurrences.

   d. Print factors and stress-occurrence array if IP(57) = 0.

   e. Call FATIGH.

   f. Calculate ultimate fatigue factor, and place in SAVE and material working area.

g. Set up stress and occurrence, scatter factor, and notch factor for endurance limit.

h. Call FATIGU.

i. Calculate ultimate fatigue factor, and place in SAVE and material working area.

j. Put material record back in library.

4. Repeat from 3 for fuselage minor frame only if material is different.

5. Print from SAVE, showing changes made in material properties by fatigue module.

## Arrays and Variables Used

ANPT, BLOCK, BMA, BMMX, BMREF, BMSMX, CON, EM, FIND, FKTFP, FKTW, FMAX, FMIN, FPT, FTU, H, IFERR, MI, NMATL, NPT, NSEG, NTOP, PCY, PMN, PMX, RA, SF, SFW, TMD, WIND

From the material record, TMD, the routine uses:

| | | |
|---|---|---|
| RA | TMD(5) | reduction in area |
| | TMD(119) | tension strain at the proportional limit |
| | TMD(121) | tension stress at the proportional limit |
| $F_{tu}$ | TMD(126) | ultimate tension stress. |

## Arrays and Variables Calculated

AN, ANPT, BMMX, E, FKT, FMAX, FMIN, HRBL, MI, NPT, NSEG, NTOP, TMD

TMD is the array used for the material record from the library. The fatigue module stores $KF_{tu}$ for each condition in all six of the possible temperature groups of material data, Table 25.

TABLE 25. FATIGUE ALLOWABLE FACTOR, $K_{F_{tu}}$, RELATIVE LOCATION IN TMD ARRAY

| Temp No. | $KF_{tu}$ Location in TMD | | | |
|---|---|---|---|---|
| | Fuselage Endurance Limit | Fuselage Pressure | SOF | WOS |
| 1 | 130 | 132 | 133 | 134 |
| 2 | 155 | 157 | 158 | 159 |
| 3 | 180 | 182 | 183 | 184 |
| 4 | 205 | 207 | 208 | 209 |
| 5 | 230 | 232 | 233 | 234 |
| 6 | 255 | 257 | 258 | 259 |

If the life calculation failed to converge, no value is stored in the $KF_{tu}$ locations for that condition; the value in the library will be used in later calculations.

MI is set up to designate if the material involved is the same as the previous one. A nonzero value indicates it is the same as previous. Used here and in FATIGU to avoid repeat calculation.

Scratch Arrays

T(1) through T(4) are used for partial values and miscellaneous temporary storage.

Labeled Common Arrays

IP, XMISC

IP(57) - print control for input factors: H, SF, FKT, FTU, E, RA, and NPT, and for the initial stress-occurrence array.

XMISC(15) - wing material number. This number is used for both SOF and WOS if material numbers are not input in fatigue input data locations 109 and/or 110.

XMISC(31) - fuselage cover material number. This number is used unless a value is input in fatigue input data location 111.

XMISC(41) - fuselage minor frame material number. This number is used unless a value is input in fatigue input data location 112.

## Mass Storage File Records Used

The material library consists of records 41 through 60. In this routine, records specified by material number plus 40 are read into the TMD array and are rewritten after insertion of the appropriate $KF_{tu}$ values.

## Error Messages

*****MATERIAL NO. XX. BEYOND YY, USED MATL. NO. 1

where XX is the input material number and YY is the number of materials in the library.

## SUBROUTINE FATIGU

## General Description

Deck name: FATIGU
Entry name: FATIGU
Called by: FTGCTL
Subroutines called: ALIFE

This routine initializes and controls the iteration for life. Material dependent factors, including points on the cyclic stress-strain and strain versus cycles to failure curves, are calculated when MI is zero (for each new material).

The following are the material dependent factors, including variable names, engineering symbols, and equations.

$$RTO = D = \ln [1.0/(1.0-R_A)] = \text{ductility}$$

$$SGF = \sigma_f = F_{tu}\,(1.0+D) = \text{fracture stress}$$

$$C1 = C_1 = 1.12\,(F_{tu}/E)\left(\sigma_f/F_{tu}\right)^{0.893}$$

$$BETA = \beta = 0.0792 + 0.179\,\log.(\sigma_f/F_{tu})$$

$$GAMMA = \gamma = \log.\left\{\frac{3.31\ D^{1/4}}{\left[1.0 - 81.4\,(F_{tu}/E)\quad \sigma_f/F_{tu}\quad 0.179\right]^{1/3}}\right\}$$

$$C2 = C_2 = 0.125\ D^{0.75}\ 10^{\gamma}$$

Other collected terms used are:

$$Q = \gamma/\beta$$

$$P = \left[(C2)\ \frac{E}{1,000}\right] / \left[(C1)\ \frac{E}{1,000}\right]^{-Q}$$

$$C1E = (C1)(E)$$

$$QP1 = Q + 1.0$$

The cyclic stress - cyclic strain equation is

$$\epsilon_t = \sigma_a/E + C_2\left(\sigma_a/C_1E\right)^{\gamma/\beta}$$

Since this equation cannot be solved directly for $\sigma_a$, nine pairs of $\sigma_a$, $\epsilon_t$ values are set up for use by the interpolation-iteration routine ACURVE which returns $\sigma_a$ for a given $\epsilon_t$.

The ln ln plot of the equation has a change of slope where the dominant term changes and the nine sigmas are calculated to cover the expected range adequately. Let $\sigma_a$ for this curve array be $y_i$.

| | | |
|---|---|---|
| First point | $y_1$ | $= 1.0$ |
| Last point | $y_9$ | $= 100{,}000{,}000$ |
| Midpoint of curved portion | $y_5$ | $= E\left(C_1^{\gamma/\beta} / C_2\right)^{\frac{1.0}{\gamma/\beta - 1.0}}$ |

$$y_4 = y_5/2.0$$

$$y_2 = \left(y_4\right)^{1/3}$$

$$y_3 = \left(y_2\right)^{2}$$

$$y_6 = y_5 + y_4$$

$$y_7 = \left(y_9/y_6\right)^{1/3} (y_6)$$

$$y_8 = \left(y_9/y_6\right)^{2/3} (y_6)$$

$\epsilon_t$ is calculated for each of these sigmas to form the x array. The natural logarithms of these x and y arrays are placed in C1X and C1Y, respectively, for use by ACURVE.

Ten points are calculated for the strain versus cycles-to-failure equation

$$\epsilon_t = C_1/N_f^{\beta} + C_2/N_f^{\gamma}.$$

The $N_f$ values used are 1.0, 10.0, 100.0, 1,000.0, 1.0 x $10^4$, 1.0 x $10^5$, 1.0 x $10^6$, 1.0 x $10^8$, 1.0 x $10^{11}$, 1.0 x $10^{15}$, 1.0 x $10^{20}$. $\varepsilon_t$ values are calculated for these $N_f$'s. The natural logarithms of $\varepsilon_t$ and $N_f$ values are stored in C2X and C2Y, respectively. These are the arrays used by the interpolation-iteration routine BCURVE.

The initial occurrences for each max-min stress are multiplied by the fraction of life per block and placed in array CYC. Also specified life, H, is multiplied by scatter factor to give hours required, HREQ.

The iteration for life is also aided by a ln-ln line estimation. This line is hours of life versus summation of the maximum stresses (FMAX). The first point is the initial stress level and the life (HCALC) that it yields. The second point is estimated by a point-slope calculation to obtain a stress level for the required life (HREQ). The initial estimate of the slope is the value in data location 64 (CON(14)).

The third and subsequent estimates are obtained by using two-point line calculation, always keeping the two points nearest the desired life value (HREQ). When a summation of FMAX has been estimated, the factor to adjust the input stress levels is the ratio of the new estimate to the previous one. Iteration proceeds until HCALC is within ±1 percent of HREQ. This tolerance is in data location 63 (CON(13)), and can of course be modified by the user.

Arrays and Variables Used

AN, ANPT, BLOCK, CON, DAM1, DAM2, E, FMAX, FMIN, FTU, H, HCALC, IP, MI, NPT, RA, SF.

Arrays and Variables Calculated

BETA, CYC, C1, C1E, C1X, C1Y, C2, C2X, C2Y, FMAX, FMIN, FTOP, GAMMA, HREQ, ND(44)[IFERR], NTIME, NTOP, P, PC1, PC2, Q, QP1, RSF, SGF.

Scratch Variables [used only in this routine and not for output, except for checkout, IP(58)]

HN, HR, H1, H2, MTIME, RE, RTO, RTO2, S, SFM, SFN, SF1, SF2, SUMF, TF.

S(42) RSF reciprocal of scatter factor

S(87) SFM ln (SUMF) initialized to first value and thereafter is the prior value.

| | | |
|---|---|---|
| S(88) | SUMF | summation of initial FMAX array. |
| S(89) | FTOP | largest initial FMAX. |
| S(90) | H1 | In (HCALC) first time, and maintained as the life coordinate of the left point. |
| S(91) | SF1 | In (SUMF) first time, and thereafter the $\Sigma f_{max}$ coordinate of the left point. |
| S(92) | H2 | second time set as life coordinate of right point, from H1 or HN |
| S(93) | SF2 | second time set as the $\Sigma f_{max}$ coordinate of the right point, from SF1 or SFN |
| S(94) | HN | In (HCALC) second time and after |
| S(95) | SFN | calculated ln ($\Sigma f_{max}$) coordinate to correspond to HR each time, to get new a FACT. |
| S(96) | HR | In (HREQ) |
| S(97) | FACT | FACT = $e^{(SFN-SFM)}$; this is the factor applied to FMAX and FMIN array values for each new iteration. |
| S(98) | TF | product of all FACT values used |
| S(99) | HREQ | hours required life |
| S(100) | HCALC | hours calculated life each pass. |
| ND(46) | MTIME | counter for repeat attempts to get positive life if first life is negative; after 5, get error print. |

Labeled Common Arrays

IP

IP(57) Final damage table (see Figure 41)

IP(58) Checkout-type print, material dependent factors, and curve points, both real and logarithmic values, are printed if the material is different from the condition before (MI=1) shown in Figure 45.

Also, a three-line print each life iteration, as in Figure 46.
NTIME, HCALC, FACT, TF
H1, SF1, H2, SF2
HN, SFN, HR, ln(FACT) first time only

## Mass Storage File Records Used

None

## Error Messages

When the life calculation fails to converge, the following printout is produced:

One of the following heading lines

1. WORKING ON WING AT SIDE OF FUSELAGE

2. WORKING ON WING OUTER PANEL STATION

3. WORKING ON FUSELAGE PRESSURE CYCLES

4. WORKING ON FUSELAGE ENDURANCE LIMIT

```
*** ERROR IN FATIGUE, PERTINENT DATA FOLLOWS...
NTIME= ____, HCALC= ____, FTU= ____, E= ____, RA= ____
HREQ= ____, ANPT= ____, SF= ____

        FMAX      FMIN      CYC
         .         .         .
         .                   .
         .         .         .
```

The arrays at the time are printed. Followed by the S array, five values per line.

S(20) through S(28), refer to ACURVE, page 343

S(30) through S(33), refer to BCURVE, page 345

S(41) HRBL hours per block

S(42) RSF reciprocal of scatter factor

S(51) through S(86), refer to ALIFE, page 339

S(87) through S(100) used in this routine, refer to descriptions under "Scratch Variables"

If the value of HCALC is zero, the trouble occurred in ALIFE in the search for $\sigma_{max}$. If there is a value in HCALC, the failure was in FATIGU. either NTIME=500, or the iteration cannot converge to HREQ. In this case, IN=H1, H2, or HR. Check the value used for tolerances D(61) and D(63).

The first line of the printing may be "NEGATIVE LIFE THRU 5 ITERATIONS". When the initial stresses are too high, the first calculated life, HCALC, can be negative. The program divides the stress by two and goes through the life calculation again until HCALC is positive, but no more than five times.

## SUBROUTINE ALIFE

### General Description

Deck name: ALIFE
Entry name: ALIFE
Called by: FATIGU
Subroutines called: ACURVE, BCURVE

This is the routine that calculates life by the strain-cycling method. The calculation procedure is as follows: To facilitate the comparison of these equations with those in the methods section, the dependent variable is given in engineering notation and program variable name.

$f_{maxj}$ = FMAX = maximum stress

$f_{minj}$ = FMIN = minimum stress

$nj$ = CYC = number of occurrences

$\sigma_{max}$ = SGMX = 1000.x for f(x) = 0

where

$$f(x) = x^2 + Px^{\gamma/\beta + 1} - (K_N f_{maxj})^2/10^6$$

$$\varepsilon_{max} = \text{EPSMX} = \sigma_{max}/E + C_2 (\sigma_{max}/C_1 E)^{\gamma/\beta}$$

$$f_1 = F1 = \frac{1}{2}(f_{max_j} - f_{min_j}) \text{ if } f_{min_j} \geq 0.0$$

$$\text{or } \frac{1}{2} f_{max_j} \text{ if } f_{min_j} < 0.0$$

$$\Delta f_2 = DF2 = f_1 - f_{min_j}$$

$$\Delta\sigma_1 = DSG1 = K_N\,(f_{max_j} - \Delta f_2)$$

$$\Delta\varepsilon_1 = DEPS1 = \Delta\sigma_1/E$$

$$\Delta\varepsilon_2 = DEPS2 = \Delta f_2\, \varepsilon_{max}/f_{max_j}$$

$\Delta\sigma_2$ = DSG2 = ACURVE $(|\Delta\varepsilon_2|)$; i.e., from cyclic stress-strain curve

$$\Delta\varepsilon_{range_j} = DEPS_j = \Delta\varepsilon_1 + \Delta\varepsilon_2$$

$$\sigma_{min} = SGMN = \sigma_{max} - \Delta\sigma_1 - \Delta\sigma_2$$

$$\sigma_{mean_j} = SGMEN_j = \frac{1}{2}(\sigma_{max} - \sigma_{min})$$

$$\varepsilon_a = EPSA = \frac{1}{2}\Delta\,\varepsilon_{range_j}$$

$$\varepsilon_t = EPST2 = \varepsilon_a/(1.0 - \sigma_{mean_j}/\sigma_f)$$

$N_1$ = TNF1 = BCURVE $(|\varepsilon_t|)$; i.e., from strain versus cycles to failure curve

$d_{1_j}$ = $DAM1_j$ = $nj/N_1$ = damage for one max-min pair for first block

SDAM1 = $\Sigma$ DAM1j

| | | $f_{max}$, $f_{min}$ opposite sign | | $f_{max}$, $f_{min}$ same sign |
|---|---|---|---|---|
| $\Delta\sigma_{res_1}$ | = | DSGR1 = $\frac{1}{2} K_N f_{min_j}$ | or | $\frac{1}{2} K_N f_{max_j}$ |
| $\Delta\varepsilon_{res_2}$ | = | DEPSR2 = $\frac{1}{2} f_{min_j} \varepsilon_{max}/f_{max_j}$ | or | $\frac{1}{2}\varepsilon_{max}$ |
| | | $S(1) = \sigma_{min}$ | or | $\sigma_{max}$ |

$\Delta\sigma_{res_2}$ = DSGR2 = ACURVE $(|\Delta\varepsilon_{res_2}|)$

$\sigma_{res_j}$ = $SGR1_j = S(1) - \Delta\sigma_{res_1} - \Delta\sigma_{res_2}$

$\Delta\varepsilon_{max}$ = DEPSMX = largest $\Delta\varepsilon_{range_j}$ (subscript value of this point is saved in JMAX)

$\sigma_{res}{}^*$ = SGR1MX = $\sigma_{res}$ that corresponds to $\Delta\varepsilon_{max}$; i.e., SGR1(JMAX)

$\bar{\sigma}_{mean}$ = SGBMEN = $\sigma_{mean_j} + \sigma_{res}{}^* - \sigma_{resj}$

$\bar{\varepsilon}_t$ = EPSBT = $\frac{1}{2} \Delta\varepsilon_{range_j}/(1.0 - \bar{\sigma}_{mean}/\sigma_f)$

$N_2$ = TNF2 = BCURVE $(|\bar{\varepsilon}_t|)$; i.e., from strain versus cycles to failure curve

$d_{2_j}$ = DAM2j = $nj/N_2$ = damage for one max-min pair for second block

SDAM2 = $\Sigma$ DAM2j

Life = HCALC = $[(RSF - \Sigma d_{1j})/\Sigma d_{2_j} + 1.0]$ HRBL

Arrays and Variables Used

CON, CYC, CIE, C2, E, FKT, FMAX, FMIN, HRBL, NPT, P, Q, QP1, RSF, SGF

Arrays and Variables Caclulated

DAM1, DAM2, HCALC

## Scratch Arrays and Variables

A, B, DEPS, DEPSMX, DEPSR2, DEPS1, DEPS2, DF2, DSGR1, DSGR2, DSG1, DSG2, EPSA, EPSBT, EPSMX, EPSTJ, EPST2, F, FA, FSQ, F1, JMAX, S(1), SDAM1, SDAM2, SGBMEN, SGMEN, SGMN, SGMX, SGR1, SGR1MX, TEPSTJ, TNF1, TNF2

S(51) through S(56) and S(59) are used in the search for sigma max ($\sigma_{max}$); refer to the foregoing equation.

| | | | |
|---|---|---|---|
| S(51) | A | S(55) | FA |
| S(52) | B | | |
| S(53) | F | S(59) | FSQ = $(K_N f_{max_j})^2/10^6$ |
| S(54) | X | | |

A and B are the bracketing values of X being used, F is the value of the function at X, and FA is the value at A.

The description of the remaining values is best found in the foregoing calculation procedure.

| | |
|---|---|
| S(56) | F1 |
| S(57) | DF2 |
| S(58) | not used |
| S(60) | through S(62) not used |
| S(63) | DSG1 |
| S(64) | DSG2 |
| S(65) | EPSA |
| S(66) | SGMN |
| S(67) | SGMX |
| S(68) | not used |
| S(69) | TNF1 |
| S(70) | TNF2 |
| S(71) | DEPS1 |
| S(72) | DEPS2 |
| S(73) | DSGR1 |
| S(74) | DSGR2 |
| S(75) | EPSBT |
| S(76) | EPSMX |
| S(77) | EPSTJ = absolute value of DEPS2 and DEPSR2 |
| S(78) | EPST2 |
| S(79) | SDAM1 |
| S(80) | SDAM2 |
| S(81) | not used |
| S(82) | DEPSMX |
| S(83) | DEPSR2 |

S(84) SGBMEN
S(85) SGR1MX
S(86) TEPSTJ = absolute value of EPST2 and EPSBT

## Labeled Common Arrays

None

## Mass Storage File Records Used

None

## Error Messages

FAILED TO FIND SGMX IN ALIFE SUBROUTINE *******
I = ______ X = ____ F = _____ A = ______ FA = ______ B = ______

This will be followed by the error print from FATIGU which is described therein. Probable cause of this failure would be errors in input fuselage pressure data.

# FUNCTION ACURVE

## General Description

Deck name: ACURVE
Entry name. ACURVE(x)
Called by: ALIFE
Subroutines called: none

This routine uses a linear interpolation and iteration scheme to evaluate the cyclic stress-strain equation and return a value of y for a given x in the following equation.

$$x = y/E + C_2 \, (y/C_1 E)^{\gamma/\beta}$$

The required constants and nine points (C1X, C1Y) of the curve are calculated in FATIGU.

The procedure followed is:

1. Linear interpolation to get y for argument x.

2. Calculate x for the interpolated y.

3. If the x calculated is within tolerance, return that y.

4. If not within tolerance, use the interpolated y and calculated x as one of the points in the linear interpolation - continuing until the calculated x is near enough to the argument x. The tolerance is in data location 62 (CON(12)).

If the argument is lower than the first point, use the y for the first point but extrapolate on the high end if necessary.

## Arrays and Variables Used

CON, CIE, C1X, C1Y, C2, E, O

The tolerance is in CON(12), data location 62.

## Arrays and Variables Calculated

Since this is a function subprogram, the result is returned directly to the equation where it appears.

## Scratch Arrays and Variables

| | | | |
|---|---|---|---|
| S(20) | XI | = | natural logarithm (ln) of the argument |
| S(21) | XCL | = | ln of XCR |
| S(22) | XJ | = | curve point lower than XI |
| S(23) | YJ | = | curve y for XJ |
| S(24) | XK | = | curve point higher than XI, or highest point if extrapolated. |
| S(25) | YK | = | curve y for XK |

S(26) YI = interpolated (ln) value for XI

S(27) YREAL = $e^{YI}$, real value of interpolated YI

S(28) XCR = value of x from stress-strain equation using y = YREAL

In subroutine area only: IFUP set to 1 when extrapolating. XPRIOR, YPRIOR set to real values of input and calculated result, respectively, so that the next argument may be checked against XPRIOR. This was set up because, when $f_{min} = 0$, the arguments $\Delta\varepsilon_2$ and $\Delta\varepsilon_{res_2}$ are the same and this arrangement avoided the duplicate search.

## Labeled Common Arrays

None

## Mass Storage File Records Used

None

## Error Messages

None

FUNCTION BCURVE

## General Description

Deck name: BCURVE
Entry name: BCURVE(x)
Called by: ALIFE
Subroutines called: none

This routine uses a linear interpolation and iteration scheme to evaluate the equation relating strain to cycles-to-failure. The routine returns the value of y for a given x for the following equation.

$$x = c_1/y^{\beta} + C_2/y^{\gamma}$$

The required constants and 10 points on the curve (C2X, C2Y) are calculated in FATIGU.

The procedure followed is:

1. Linear interpolation to get y for the argument x.

2. Calculation of x for the interpolated y.

3. If the calculated x is within the tolerance, return that y for the result. The tolerance is in data location 62 (CON(12)). The test is $|\Delta x|/x \cdot$ CON(12).

4. If not within tolerance, the point defined by x calculated and y iterated is used as one of the points in the linear interpolation and the procedure is repeated.

This curve is not extrapolated; a value beyond either and returns the end value.

## Arrays and Variables Used

BETA, CON, C1, C2, C2X, C2Y, GAMMA

Calculation tolerance is CON(12), data location 62.

## Arrays and Variables Calculated

Since this is a function subprogram, the result is returned directly to the equation where it appears.

## Scratch Arrays and Variables

| | | | |
|---|---|---|---|
| S(30) | XI | = | natural logarithm (ln) of the argument |
| S(31) | XCL | = | ln of XCR |
| S(32) | XJ | = | x of curve higher than XI |
| S(33) | XK | = | y of curve corresponding to XJ |

| | | | |
|---|---|---|---|
| S(34) | XK | = | x of curve point lower than XI |
| S(35) | YK | = | y of curve corresponding to XK |
| S(36) | YI | = | interpolated value for XI |
| S(37) | YREAL | = | $e^{YI}$, real value of interpolated y |
| S(38) | XCR | = | value of x from the equation using y = YREAL. It is this value that is tested against the argument for agreement. |

Note that, in this curve, x decreases for increasing y.

## Labeled Common Arrays

None

## Mass Storage File Records Used

None

## Error Messages

None

## Section IV

## MODULE FLOW CHARTS AND FORTRAN LISTS

### FLOW CHART USAGE

The automatically generated computer program flow charts (AUTOFLOW) presented in this document include a table of contents, flow charts, and FORTRAN lists of all routines in the module. The 80-column card lists are sequenced and grouped by routine.

Because the AUTOFLOW system used is IBM-oriented, the functions of the BUFFERIN and BUFFEROUT statements are not recognized, but these statements appear in proper order in note boxes. Also, the PROGRAM name does not appear on the main program, and library routines READMS and WRITMS are listed as undefined external references.

### CROSS-REFERENCE LIST

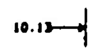
The AUTOFLOW table of contents which precedes the flow charts and FORTRAN lists serves to cross reference the latter two. This table lists the following from left to right:

- The card identification from columns 73 through 80 of this card, or card sequence number. When sequence number is used in place of card identification, it is enclosed in parentheses.
- The page and box number where this card is displayed in a flow chart.
- The FORTRAN statement number from columns 1 through 5 of this card.
- The card identification(s) or sequence number(s) of the card(s) referring to this card (repeated as required).
- The pages and box numbers where the cards referring to this card are displayed in a flow chart (repeated as required).

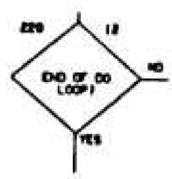
### FLOW CHARTS

The flow charts produced by AUTOFLOW use USASI conventional symbols. Since the flow charts are mechanically drawn from the program source deck, there are no omissions or vague generalizations about the processing within the boxes.

Every box on each page is uniquely numbered and may be referred to from elsewhere in the program. The source of a reference to a box will be indicated by showing the page and box number. If the number is followed by an asterisk, there are multiple references to this point, and the others may be found by using the cross-reference list.

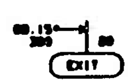


The most-often-used symbol is the decision box. Like all boxes, its box number is above and to the right of the box. Its FORTRAN statement number is above and to the left of the box. The decision choices for the paths are printed.

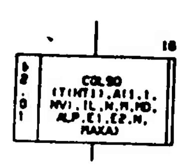


The unconditional transfer connecter has its page number destination printed above or to the left of the box number destination within the connecter. If there is a FORTRAN statement number at the destination, it is printed below the connecter.

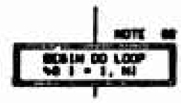
The exit box example shows a connecter from page 9, box 15.



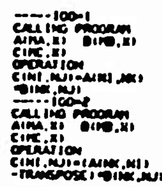

The subroutine call box includes the calling sequence. The page and box numbers of the flow chart of the called subroutine are shown on the left-hand side of the box. The page number is above the box number.



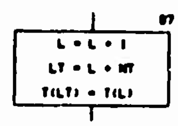
The note box encloses comments of a functional nature,



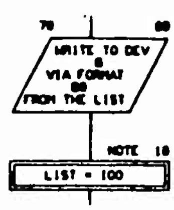

as differentiated from the 21 column comments, which are left justified without a box, that show the comment cards included in the FORTRAN deck.



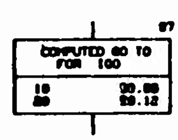

The process box is used to enclose FORTRAN arithmetic statements.




Input and output are shown as communicating with a device. The list used follows, if appropriate:




The computed GØ TØ becomes a branch table showing the page and box number of each of the ordered branches.




The column connecters and initial connecters are the only boxes without external box numbers. The function of the initial connecter is always clear,

but the label given is the symbol in the next FORTRAN card, which is often blank.

The column connecter identifies the page and box number to which it connects.

TABLE OF CONTENTS

FOR

AUTOFLOW CHART SET

```
CARD ID   PAGE/BOX   NAME                       REFERENCES  (SOURCE SEQUENCE NO. AND PAGE/BOX)


FORTRAN MODULE    FATIGUE MODULE


CHART TITLE - INTRODUCTORY COMMENTS


CHART TITLE - PROCEDURES

(000017)   2.02  10
(000017)   2.02             (000017)   2.03
(000022)   2.06  20
(000023)   2.07  22         (000021)   2.05
(000024)   2.08  23
(000025)   2.09  24         (000023)   2.07
(000026)   2.10  25
(000027)   2.11  26         (000025)   2.09
(000038)   2.17  5001
(000045)   2.19  150
(000053)   2.25  5002       (000037)   2.16

CHART TITLE - NON-PROCEDURAL STATEMENTS


CHART TITLE - INTRODUCTORY COMMENTS


CHART TITLE - FUNCTION  ACURVE(X)

(000087)   5.05  10         (000085)   5.04    (000090)   5.07    (000095)   6.03
(000089)   5.06  15         (000085)   5.04
(000091)   5.08  20
(000093)   6.01  30         (000090)   5.07
(000094)   6.02             (000098)   6.04
(000096)   6.04  35
(000097)   6.05  40         (000092.   5.08    (000095)   6.03
(000101)   6.08  45         (000122)   6.16    (000115)   6.17    (000112)   6.18
(000108)   6.09  50
(000109)   6.10  55
(000124)   6.11  998        (000107)   6.08    (000109)   6.10    (000116)   6.14
(000125)   6.12  999        (000081)   5.02    (000088)   5.05
(000116)   6.14  70         (000108)   6.09
(000117)   6.15  72
(000118)   6.16  75         (000116)   6.14
(000113)   6.17  65         (000109)   6.10
(000110)   6.18  60

CHART TITLE - NON-PROCEDURAL STATEMENTS


CHART TITLE - INTRODUCTORY COMMENTS


CHART TITLE - SUBROUTINE  ALIFE

(000135)   9.01  ALIFE      (000501)  18.12-X
(000173)   9.03             (000248)  10.23
(000177)   9.04  90         (000183)   9.08
(000180)   9.06  95
(000184)   9.09  97         (000179)   9.05
(000185)   9.10  100        (000195)   9.19    (000198)   9.20
(000190)   9.13  110
(000200)   9.14  166        (000181)   9.07    (000185)   9.10
(000191)   9.16  120        (000189)   9.12
(000192)   9.17  130        (000190)   9.13
(000193)   9.18  140
(000194)   9.19  150
(000198)   9.20  160        (000192)   9.17
(000208)  10.01  170        (000179)   9.05    (000189)   9.12    (000190)   9.13    (000191)   9.16    (000192)   9.17
                            (000193)   9.18
(000210)  10.04             (000209)  10.02
(000211)  10.06             (000210)  10.04
(000219)  10.10             (000218)  10.08
(000238)  10.16  200        (000230)  10.13    (000231)  10.14
(000239)  10.17  210        (000235)  10.15
(000242)  10.20             (000241)  10.18
(000248)  10.23  250        (000245)  10.21
```

```
CARD ID   PAGE/BOX   NAME              REFERENCES  (SOURCE SEQUENCE NO. AND PAGE/BOX)

(000255)  10.26                 (000283)  10.28
(000263)  10.28  300
(000268)  10.30  999            (000204)   9.15

CHART TITLE - NON-PROCEDURAL STATEMENTS


CHART TITLE - INTRODUCTORY COMMENTS


CHART TITLE - FUNCTION  BCURVE(X)

(000299)  13.03  10             (000302)  13.05    (000308)  13.08
(000301)  13.04  20             (000298)  13.02
(000304)  13.06  30             (000302)  13.05
(000375)  13.07                 (000307)  13.09
(000307)  13.09  35
(000308)  13.1   40             (000306)  13.08
(000313)  13.11  45             (000323)  13.15    (000326)  13.18
(000320)  13.14  50
(000321)  13.15  55
(000327)  13.16  999            (000300)  13.03    (000319)  13.13    (000320)  13.14
(000324)  13.18  60             (000320)  13.14

CHART TITLE - NON-PROCEDURAL STATEMENTS


CHART TITLE - INTRODUCTORY COMMENTS


CHART TITLE - SUBROUTINE  FATIOU

(000338)  16.01  FATIOU         (000845)  26.33-X  (000931)  29.20-X  (000955)  29.36-X
(000400)  16.09  5001
(000403)  16.12  17             (000399)  16.08
(000436)  16.17                 (000437)  16.18
(000437)  16.18  30
(000441)  16.21                 (000442)  16.22
(000442)  16.22  32
(000446)  16.25                 (000448)  16.26
(000448)  16.25  35
(000450)  16.28                 (000451)  16.29
(000451)  16.29  38
(000454)  16.31  5003
(000459)  16.38  42             (000453)  16.30
(000462)  17.01                 (000466)  17.02
(000466)  17.02  45
(000471)  17.05  5005
(000476)  17.10  49             (000381)  16.02    (000470)  17.04
(000815)  18.01  800            (000503)  18.13    (000602)  18.14    (000606)  18.16    (000546)  19.22    (000576)  19.25
                                (000504)  20.19    (000505)  20.20    (000598)  20.21
(000816)  18.02  801            (000815)  18.01
(000486)  18.04                 (000491)  18.07
(000491)  18.07  100            (000488)  18.05
(000494)  18.08  110
(000494)  18.09                 (000494)  18.10
(000501)  18.12  150            (000562)  20.07    (000571)  20.13
(000602)  18.14  500            (000503)  18.13
(000607)  18.17  510            (000603)  18.15
(000523)  19.01  220
(000504)  19.10  160
(000505)  19.11  200
(000507)  19.13  5007
(000511)  19.16                 (000513)  19.17
(000513)  19.17  210
(000521)  19.20                 (000523)  19.11
(000545)  19.21  300            (000504)  19.10
(000547)  19.23  301
(000574)  19.24  350            (000547)  19.23
(000577)  19.26  360            (000598)  20.21    (000599)  20.22
(000550)  20.01  305            (000547)  19.23
(000557)  20.03  320            (000609)  18.17    (000588)  20.17
(000558)  20.04                 (000560)  20.05
(000560)  20.05  325
(000563)  20.08  5009
```

| CARD ID | PAGE/BOX | NAME | REFERENCES (SOURCE SEQUENCE NO. AND PAGE/BOX) | | |
|---|---|---|---|---|---|
| (000504) | 20.09 | 5010 | | | |
| (000566) | 20.10 | 5011 | (000563) 20.08 | | |
| (000580) | 20.14 | 365 | (000578) 19.25 | (000594) 20.19 | |
| (000582) | 20.15 | 367 | (000598) 20.21 | | |
| (000584) | 20.16 | 370 | (000579) 19.26 | | |
| (000592) | 20.18 | 400 | (000547) 19.23 | | |
| (000595) | 20.20 | 405 | | | |
| (000596) | 20.21 | 410 | | | |
| (00/597) | 20.22 | 420 | (000595) 20.20 | | |
| (000619) | 21.01 | 803 | (000615) 18.01 | | |
| (000622) | 21.02 | 805 | (000615) 18.01 | | |
| (000625) | 21.03 | 807 | (000615) 18.01 | | |
| (000627) | 21.04 | 809 | (000618) 18.02 | (000621) 21.01 | (000624) 21.02 |
| (000636) | 21.09 | 999 | (000543) 19.09 | (000508) 19.12 | |

CHART TITLE - NON-PROCEDURAL STATEMENTS

CHART TITLE - INTRODUCTORY COMMENTS

CHART TITLE - SUBROUTINE FTOCTL

| | | | | | |
|---|---|---|---|---|---|
| (000672) | 24.01 | FTOCTL | (000055) 2.26-X | | |
| (000687) | 24.02 | 10 | | | |
| (000687) | 24.02 | | (000687) 24.03 | | |
| (000702) | 24.05 | 100 | | | |
| (000704) | 24.06 | 101 | (000701) 24.04 | | |
| (000705) | 24.07 | 105 | (000701) 24.04 | (000703) 24.05 | |
| (000706) | 24.08 | 106 | | | |
| (000708) | 24.09 | 107 | (000705) 24.07 | | |
| (000709) | 24.10 | 110 | (000705) 24.07 | (000707) 24.08 | |
| (000710) | 24.11 | 111 | | | |
| (000712) | 24.12 | 112 | (000709) 24.10 | | |
| (000713) | 24.13 | 113 | (000709) 24.10 | (000711) 24.11 | |
| (000714) | 24.14 | 114 | | | |
| (000716) | 25.01 | 115 | (000713) 24.13 | | |
| (000717) | 25.02 | 116 | (000713) 24.13 | (000715) 24.14 | |
| (000721) | 25.04 | 200 | | | |
| (000740) | 25.06 | 201 | (000864) 27.14 | | |
| (000742) | 25.08 | 2011 | | | |
| (000743) | 25.09 | 2012 | (000741) 25.07 | | |
| (000748) | 25.12 | 2015 | (000742) 25.08 | | |
| (000758) | 25.18 | 202 | (000866) 27.15 | | |
| (000763) | 25.19 | | (000802) 26.05 | | |
| (000767) | 25.21 | | (000799) 26.03 | | |
| (000780) | 25.29 | 203 | | | |
| (000784) | 25.30 | 204 | (000779) 25.28 | | |
| (000787) | 25.31 | 205 | (000783) 25.29 | | |
| (000788) | 25.32 | 206 | (000778) 25.27 | (000779) 25.28 | |
| (000791) | 25.35 | 207 | | | |
| (000795) | 26.01 | 208 | (000790) 25.34 | | |
| (000798) | 26.02 | 209 | (000794) 25.35 | | |
| (000799) | 26.03 | 210 | (000770) 25.23 | (000789) 25.33 | (000790) 25.34 |
| (000802) | 26.05 | 212 | | | |
| (000812) | 26.09 | 5021 | | | |
| (000822) | 26.14 | 215 | (000818) 26.12 | | |
| (000824) | 26.15 | 220 | (000821) 26.13 | | |
| (000826) | 26.16 | 5002 | (000811) 26.08 | | |
| (000830) | 26.19 | | (000831) 26.21 | | |
| (000831) | 26.21 | 222 | | | |
| (000831) | 26.21 | | (000830) 26.19 | | |
| (000835) | 26.24 | | (000838) 26.26 | | |
| (000836) | 26.25 | 225 | | | |
| (000838) | 26.28 | 5003 | | | |
| (000843) | 26.32 | 5004 | (000837) 26.27 | | |
| (000848) | 26.35 | 243 | | | |
| (000850) | 27.01 | 250 | (000847) 26.34 | | |
| (000851) | 27.02 | 251 | | | |
| (000852) | 27.03 | | (000853) 27.04 | | |
| (000853) | 27.04 | 255 | | | |
| (000855) | 27.06 | 260 | (000849) 26.35 | | |
| (000856) | 27.07 | 5007 | | | |

| CARD ID | PAGE/BOX | NAME | REFERENCES (SOURCE SEQUENCE NO. AND PAGE/BOX) | | |
|---|---|---|---|---|---|
| (000859) | 27.09 | 5008 | (000855) 27.06 | | |
| (000862) | 27.12 | 266 | (000721) 25.04 | | |
| (000881) | 28.01 | 503 | (000878) 28.10 | (000973) 30.02 | (000976) 30.05 |
| (000871) | 28.04 | 500 | (000861) 27.11 | (000862) 27.12 | |
| (000876) | 28.08 | 5001 | (000873) 28.06 | | |
| (000878) | 28.10 | 501 | | | |
| (000879) | 28 11 | 502 | (000877) 28.09 | | |
| (000888) | 28.14 | 507 | (000874) 28.07 | | |
| (000890) | 28.15 | 510 | (000887) 28.03 | | |
| (000900) | 28.19 | 5005 | | | |
| (000905) | 29.01 | 5052 | (000901) 28.21 | | |
| (000907) | 29.02 | 5006 | (000899) 28.18 | (000904) 28.22 | |
| (000914) | 29.05 | | (000918) 29.08 | | |
| (000918) | 29.08 | 520 | | | |
| (000918) | 29.08 | | (000917) 29.06 | | |
| (000921) | 29.11 | | (000923) 29.12 | | |
| (000923) | 29.12 | 525 | | | |
| (000925) | 29.14 | 5009 | | | |
| (000929) | 29.18 | 5010 | (000924) 29.13 | | |
| (000933) | 29.22 | 580 | | | |
| (000935) | 29.23 | 600 | (000932) 29.21 | | |
| (000936) | 29.24 | 610 | | | |
| (000937) | 29.25 | | (000938) 29.26 | | |
| (000938) | 29.26 | 615 | | | |
| (000941) | 29.28 | 617 | (000934) 29.22 | | |
| (000950) | 29.31 | 5011 | | | |
| (000953) | 29.34 | 5012 | (000949) 29.30 | | |
| (000957) | 29.38 | 650 | | | |
| (000973) | 30.02 | 801 | | | |
| (000974) | 30.03 | 802 | (000972) 30.01 | | |
| (000959) | 30.06 | 660 | (000956) 29.37 | | |
| (000960) | 30.07 | 665 | | | |
| (000961) | 30.08 | | (000962) 30.09 | | |
| (000962) | 30.09 | 670 | | | |
| (000964) | 30.11 | 675 | (000958) 29.38 | | |
| (000968) | 30.14 | 800 | (000871) 28.04 | | |
| (000978) | 30.17 | 820 | (000988) 30.15 | | |
| (000979) | 30.18 | | (000980) 30.19 | | |
| (000980) | 30.19 | 822 | | | |
| (000982) | 30.20 | 999 | (000967) 30.13 | (000968) 30.14 | |
| (000984) | 30.21 | 5013 | | | |
| (000998) | 30.23 | 5014 | | | |

CHART TITLE - NON-PROCEDURAL STATEMENTS

| CARD ID | PAGE LOC | |
|---|---|---|
| (000019) | 2.04 | UNDEFINED - 'READMS' EXTERNAL REFERENCE |
| (000034) | 2.15 | UNDEFINED - 'READMS' EXTERNAL REFERENCE |
| (000749) | 25.13 | UNDEFINED - 'READMS' EXTERNAL REFERENCE |
| (000860) | 27.10 | UNDEFINED - 'WRITMS' EXTERNAL REFERENCE |
| (000882) | 28.02 | UNDEFINED - 'READMS' EXTERNAL REFERENCE |
| (000965) | 30.12 | UNDEFINED - 'WRITMS' EXTERNAL REFERENCE |

PROGRAM FLOW CHARTS

OF

FATIGUE MODULE

CHART TITLE - PROCEDURES

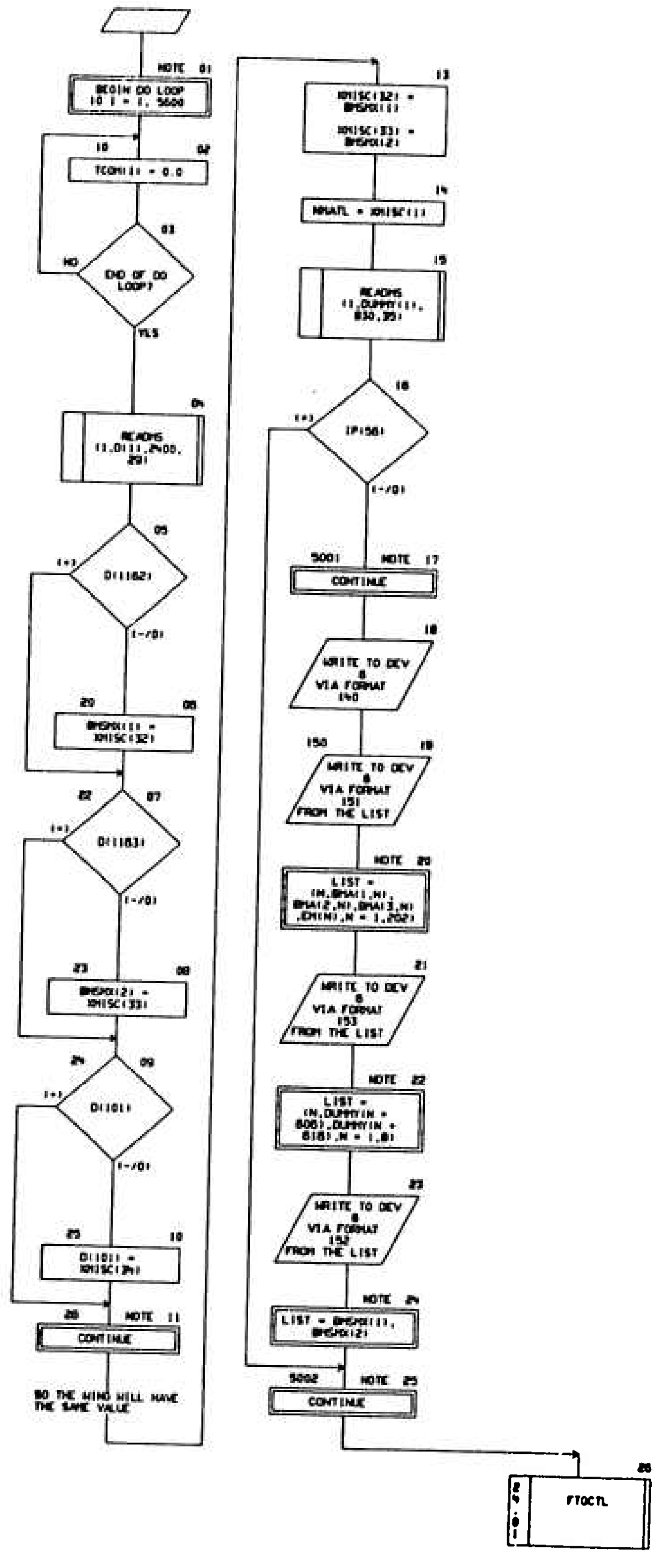


CHART TITLE - NON-PROCEDURAL STATEMENTS

```
          PROGRAM FATGUE
          COMMON TCOM(5600)
          COMMON /MISC/XMISC(100)
          COMMON /IPRINT/IP(80)
          DIMENSION D(2400), T(3000), ND(200)
          , BMA(3,220), DUMMY(830), EM(220), BMREF(20), BMSMX(2)
          EQUIVALENCE (D(1),TCOM(1)), (T(1),TCOM(2401)), (ND(1),TCOM(5401))
          , (BMA(1,1),DUMMY(1),D(1501)), (BMREF(1),D(2107)), (EM(1),D(2187))
          , (BMSMX(1),D(1182))
          , (ND(59),NMATL)
140       FORMAT(1H1,70H*** BMA(3,220) FROM SUBROUTINE FATMB IN LOADS PROGRA
          M IN RECORD 35 ***,10X,21H** FATGUE - IP(58) **//
            8X, 12H60F BEND MOM, 3X, 12H40S BEND MOM, 2X,
            18HEXCEEDANCES-GUST, 2X, 18HEXCEEDANCES-MANU  )
151       FORMAT( I15,2F15.0,2E18. 5)
153       FORMAT( 39HOREFERENCE BENDING MOMENTS FOR MANEUVER / 4X, 7HSEGMENT
           8X, 3H60F, 10X, 3H40S /(I19, 2F15.0 ) )
152       FORMAT(//10X,10HBMSMX(1) =,F15.0/10X,10HBMSMX(2) =,F15.0//)
```

FUNCTION ACURVE

CHART TITLE - FUNCTION ACURVE(X)

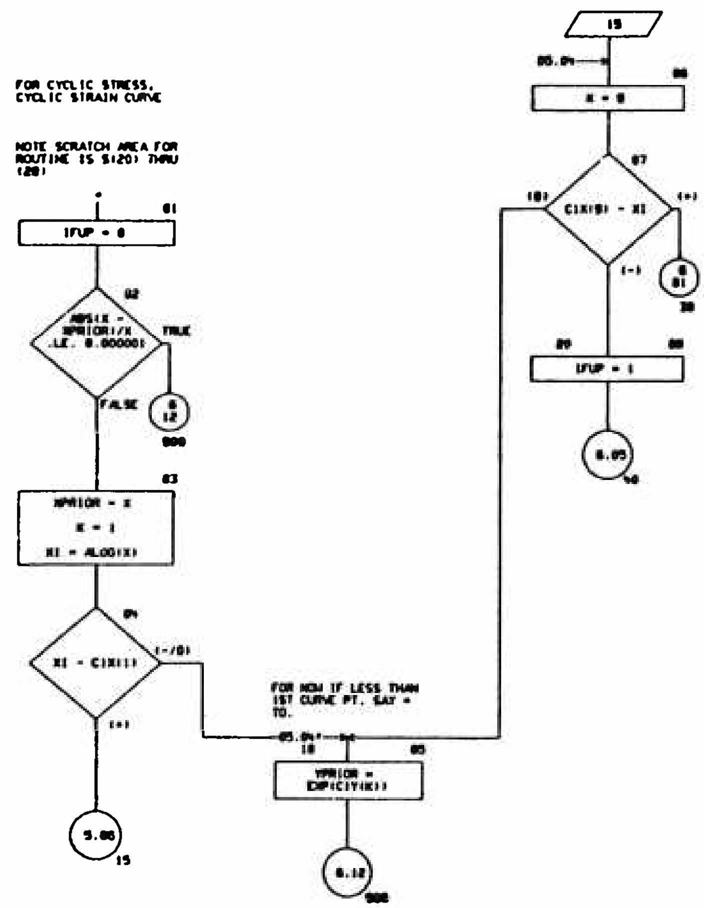


CHART TITLE - FUNCTION ACURVE(X)

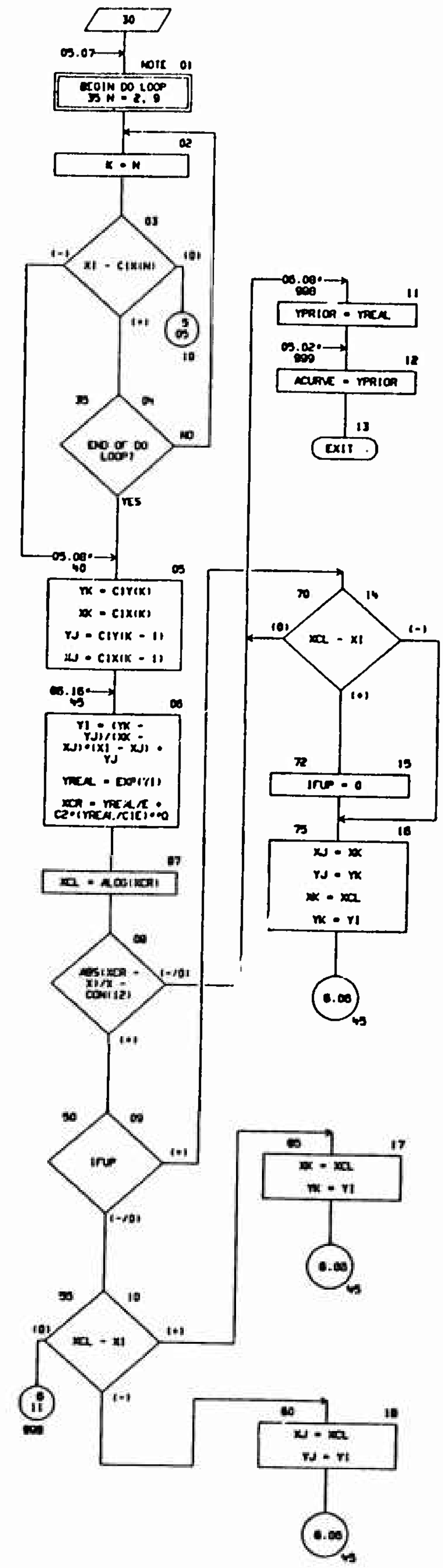


CHART TITLE - NON-PROCEDURAL STATEMENTS

```
COMMON TCOM(5600)
DIMENSION D(2400), T(3000), ND(200)
. CON(50), S(100)
. CIX(10), CIY(10)
EQUIVALENCE (D(1),TCOM(1)), (T(1),TCOM(2401)), (ND(1),TCOM(5401))
. (D(51),CON(1)), (D(105),E)
. (T(1),S(1))
. (T(1001),CIX(1)), (T(1011),CIY(1))
.                (CON(20),C2), (CON(23),Q), (CON(26),CIE)
EQUIVALENCE (S(20),XI), (S(21),XCL), (S(22),XJ), (S(23),YJ)
. (S(24),XK), (S(25),YK), (S(26),YI), (S(27),YREAL), (S(28),XCR)
DATA XPRIOR/0.0/, YPRIOR/0.0/
```

CHART TITLE - INTRODUCTORY COMMENTS

```
))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))
                          SUBROUTINE ALIFE
((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((
```

CHART TITLE - SUBROUTINE ALIFE

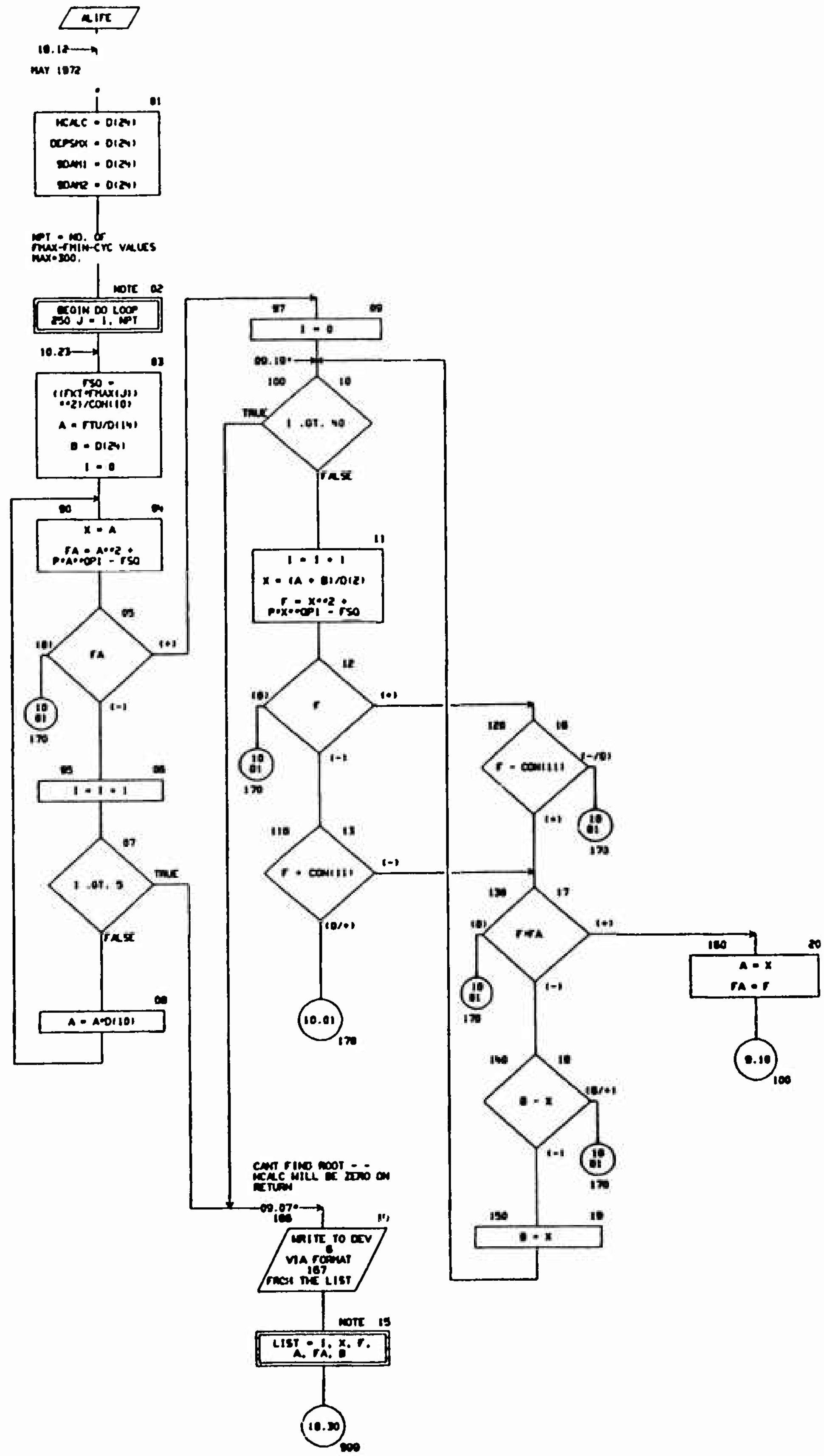


CHART TITLE - SUBROUTINE ALIFE

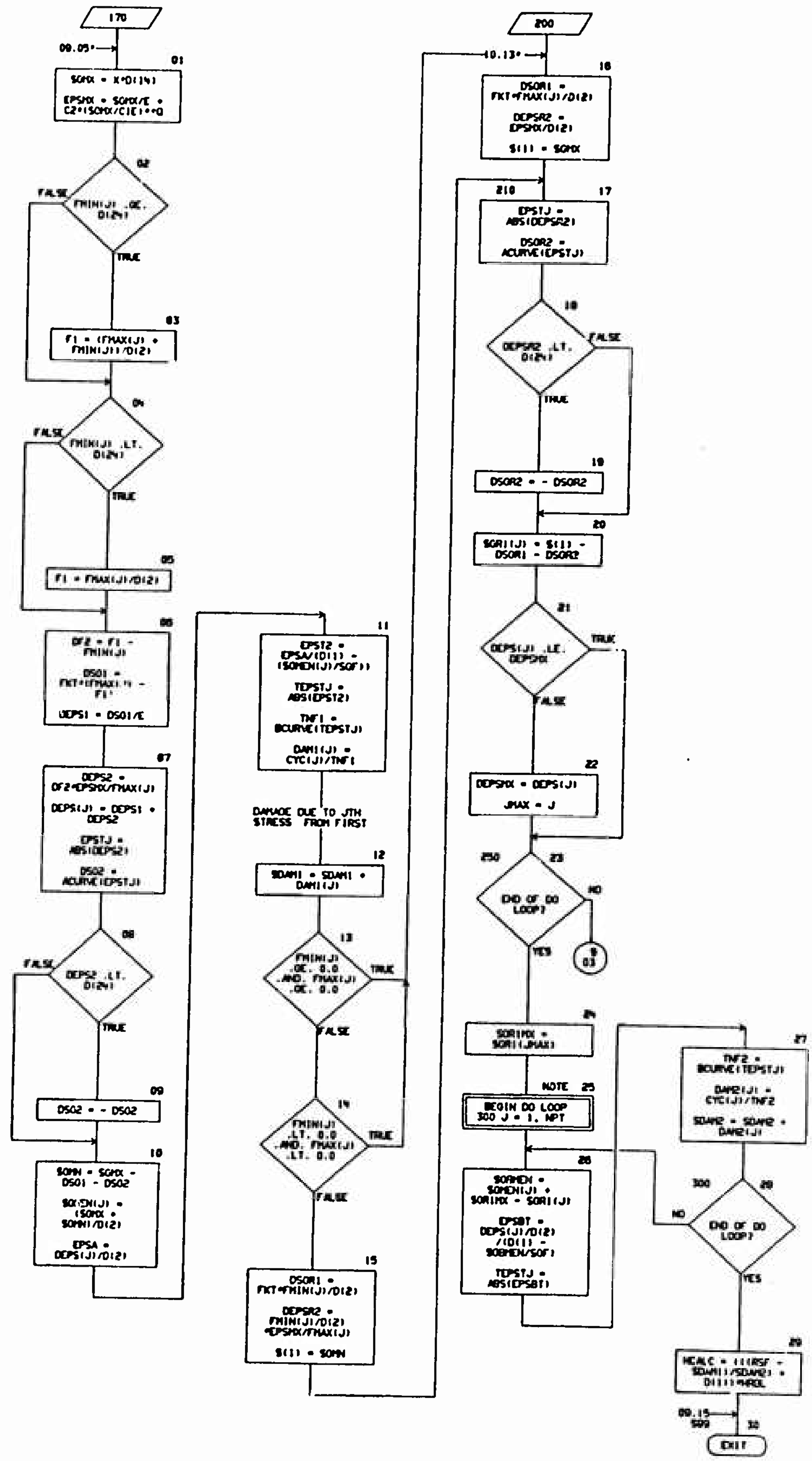


CHART TITLE - NON-PROCEDURAL STATEMENTS

```
      COMMON TCOM(5600)
      DIMENSION D(2400), T(3000), NO(200)
     , CON(30), S(100), FMAX(300), FMIN(300),          CYC(300)
     , DEPS(300), SGMEN(300), SOR1(300), DAM1(300), DAM2(300)
      EQUIVALENCE (D(1),TCOM(1)), (T(1),TCOM(2401)), (NO(1),TCOM(5401))
     , (D(51),CON(1)),                          (D(103),FKT)
     , (D(104),FTU), (D(105),E)
     , (D(201),FMAX(1)), (D(501),FMIN(1))
     , (T(1),S(1)), (T(101),CYC(1)), (T(99),NREC0), (T(100),NCALC)
     , (T(1401),DAM1(1)), (T(1701),DAM2(1))
     , (T(2001),DEPS(1)), (T(2301),SOR1(1)), (T(2601),SGMEN(1))
     ,              (CON(18),SOF), (CON(20),C2)
     , (CON(23),Q),              (CON(25),P), (CON(26),CIE)
     , (CON(27),QP1)
     ,(NO(40),NPT), (NO(41),JMAX)
      EQUIVALENCE (S(41),NRBL), (S(42),RSF)
      EQUIVALENCE (S(51),A), (S(52),B), (S(53),F), (S(54),X), (S(55),FA)
     , (S(56),F1), (S(57),DF2),             (S(59),FSQ)
     ,                            (S(63),DSO1), (S(64),DSO2)
     , (S(65),EPSA), (S(66),SOMN), (S(67),SOMX)
     , (S(69),TNF1), (S(70),TNF2), (S(71),DEPS1), (S(72),DEPS2)
     , (S(73),DSOR1), (S(74),DSOR2), (S(75),EPSBT), (S(76),EPSMX)
     , (S(77),EPSTJ), (S(78),EPST2), (S(79),SDAM1), (S(80),SDAM2)
     ,              (S(82),DEPSMX), (S(83),DEPSR2), (S(84),SGBMEN)
     , (S(85),SOR1MX), (S(88),TEPSTJ)
187   FORMAT(1H1,47HFAILED TO FIND SOMX IN ALIFE SUBROUTINE ******* //
      3H I=,I13, 2X,2HX=,1E14.8,2X,2HF=,1E14.8,2X,2HA=,1E14.8,2X,3HFA=,
      1E14.8,2X,2HB=,1E14.8 )
```

))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))

FUNCTION BCURVE

((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((

CHART TITLE - FUNCTION BCURVE(X)

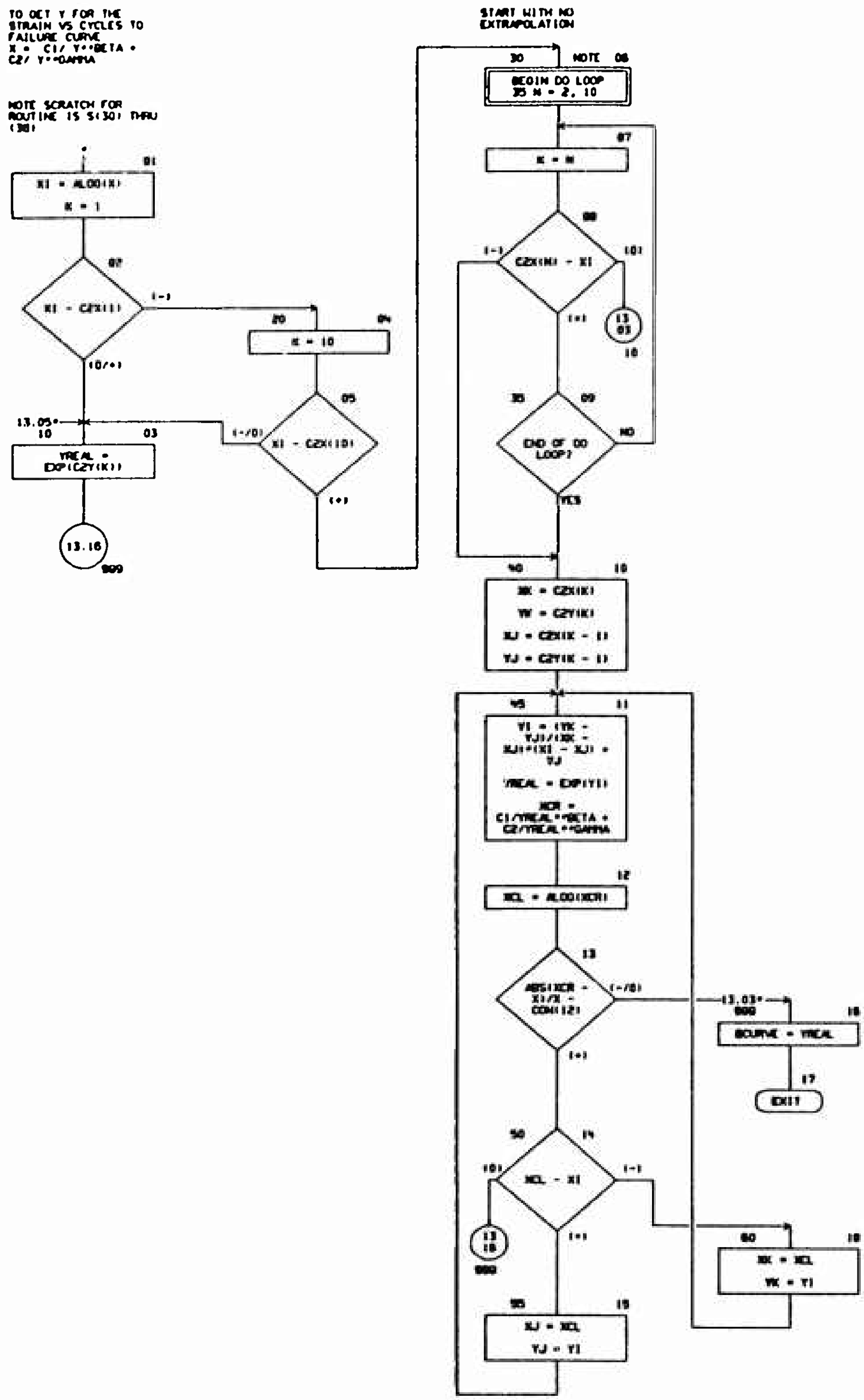


CHART TITLE - NON-PROCEDURAL STATEMENTS

```
COMMON TCOM(5600)
DIMENSION D(2400), T(3000), ND(200)
. CON(50), S(100)
.                  C2X(10), C2Y(10)
EQUIVALENCE (D(1),TCOM(1)), (T(1),TCOM(2401)), (ND(1),TCOM(5401))
. (D(51),CON(1))
. (T(1),S(1))
. (T(1021),C2X(1)), (T(1031),C2Y(1))
. (CON(19),C1), (CON(20),C2), (CON(21),BETA), (CON(22),GAMMA)
EQUIVALENCE (S(30),XI), (S(31),XCL), (S(32),XJ), (S(33),YJ)
. (S(34),XK), (S(35),YK), (S(36),YI), (S(37),YREAL), (S(38),XCR)
```

CHART TITLE - INTRODUCTORY COMMENTS

```
))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))
                          SUBROUTINE FATIOU
((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((
```

CHART TITLE - SUBROUTINE FATIGU

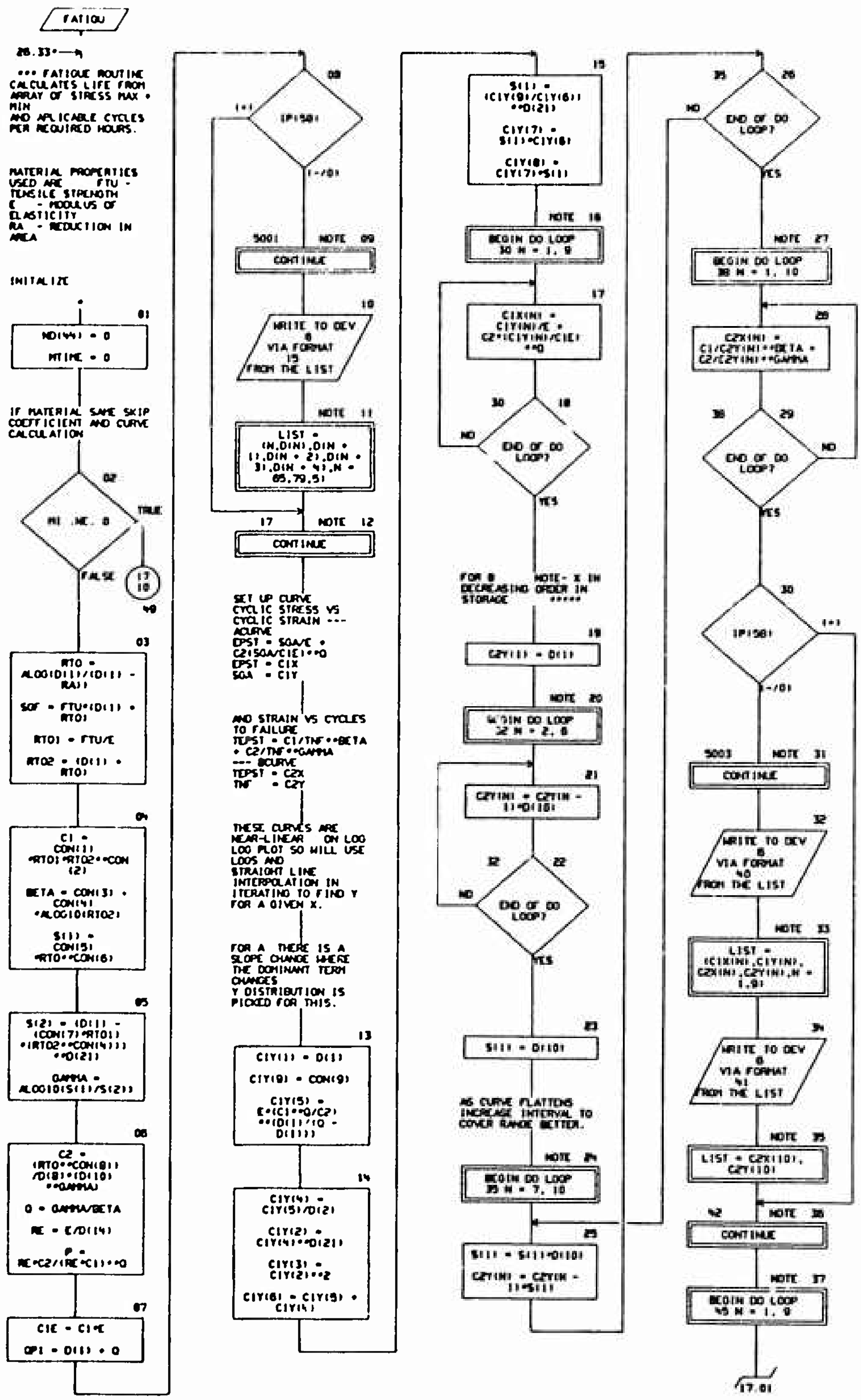


CHART TITLE - SUBROUTINE FATIGU

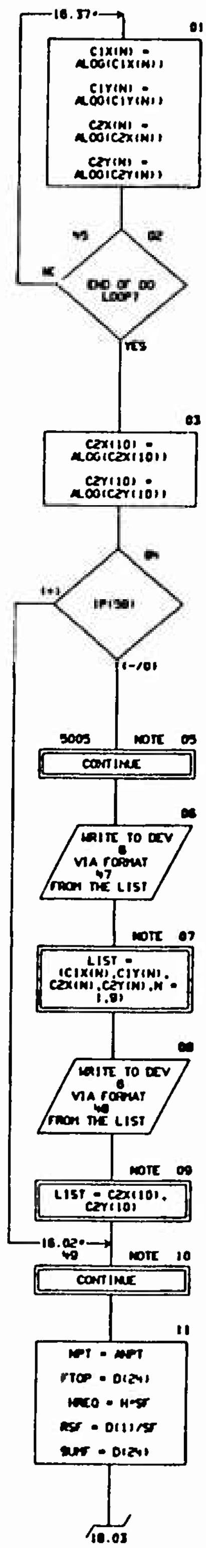


CHART TITLE - SUBROUTINE FATIGU

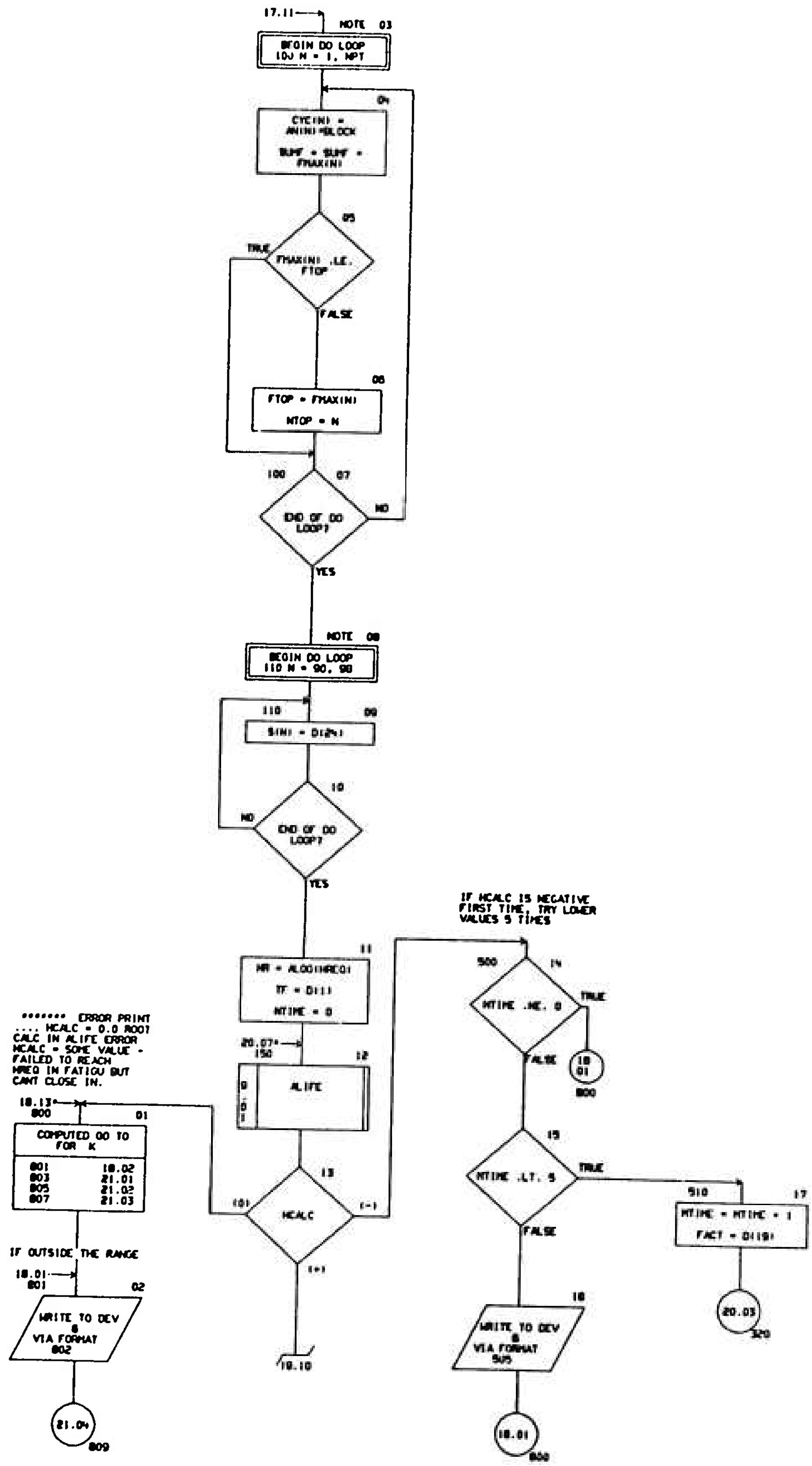


CHART TITLE - SUBROUTINE FATIGU

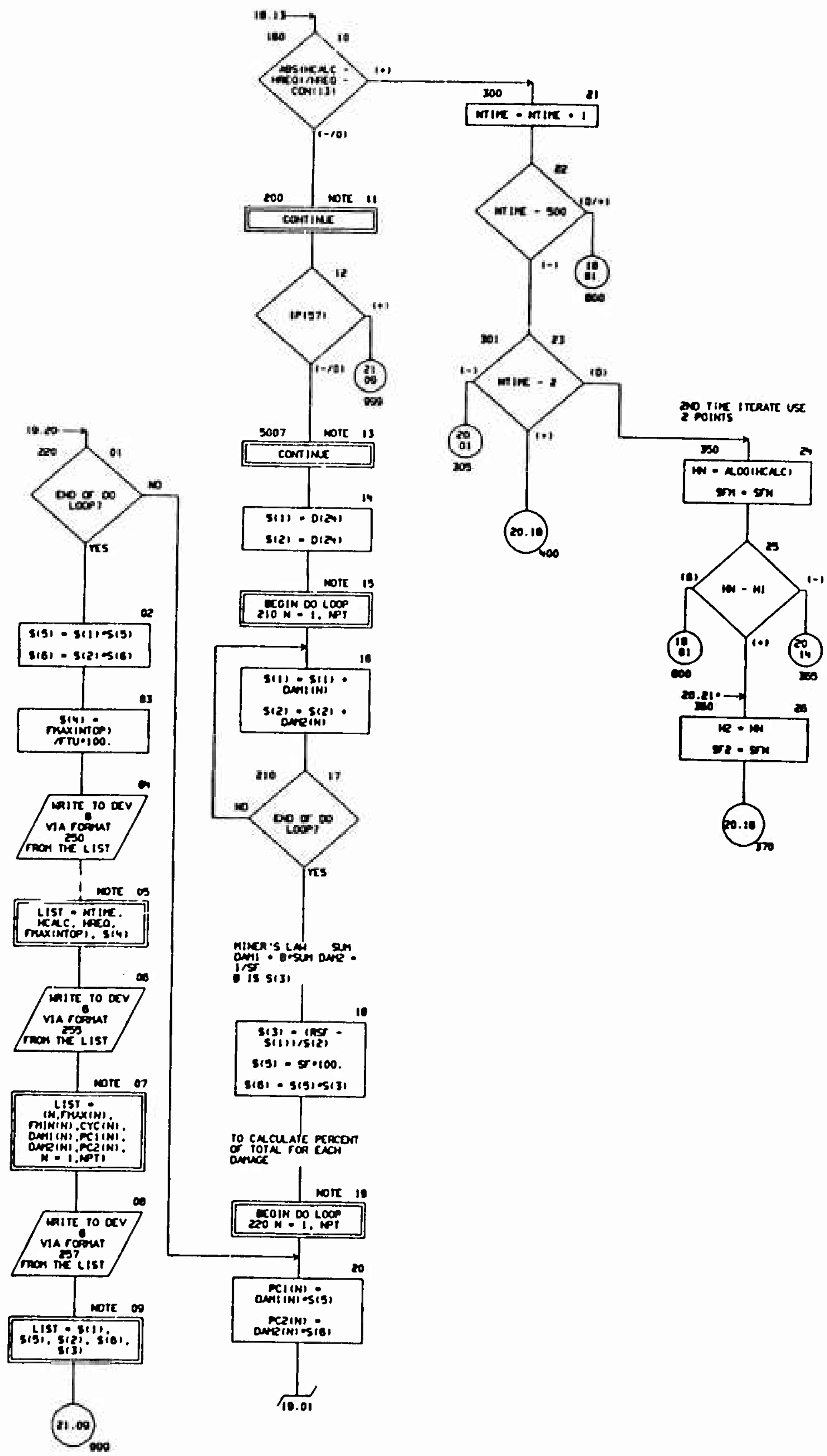


CHART TITLE - SUBROUTINE FATIOU

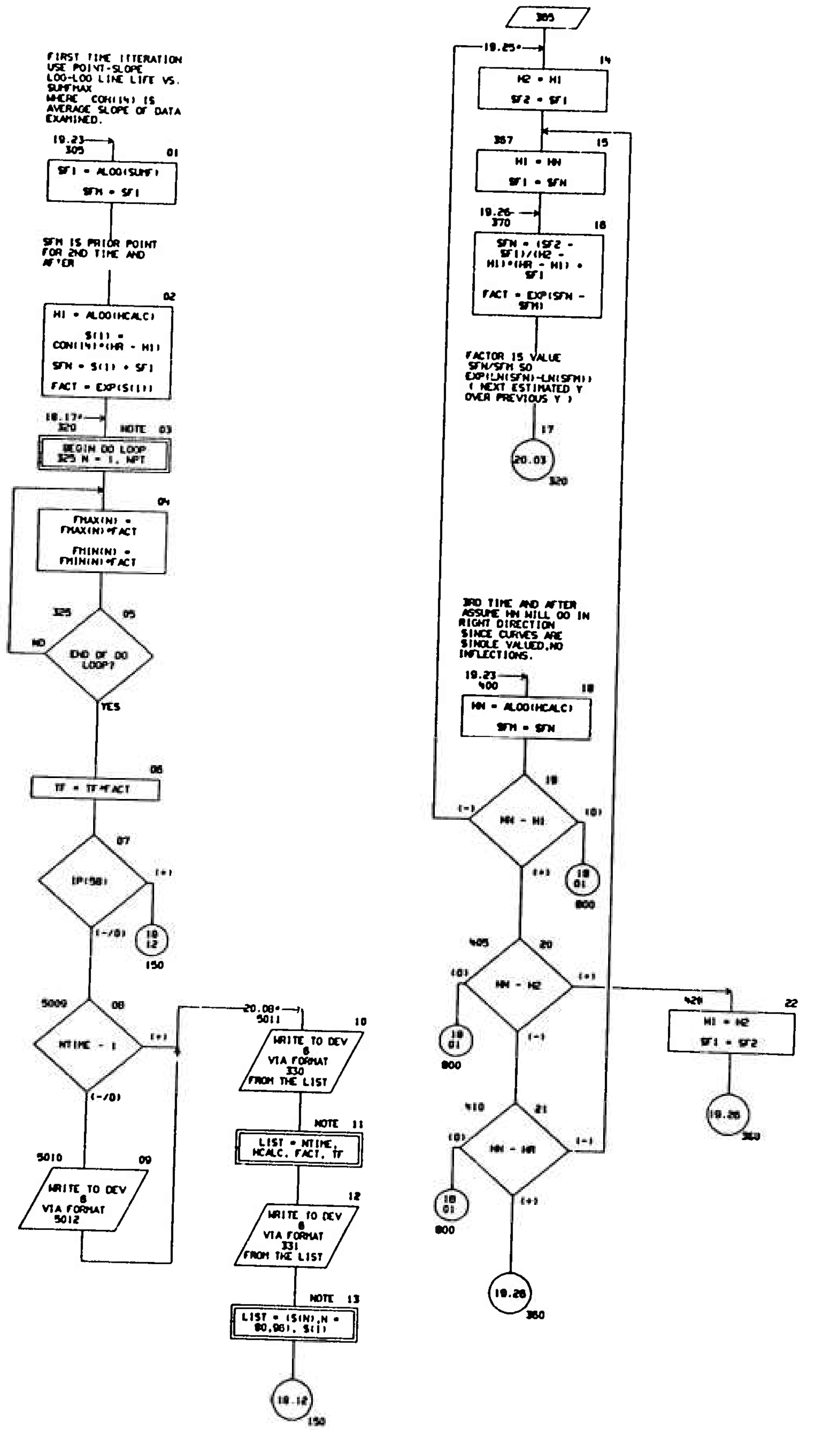


CHART TITLE - SUBROUTINE FATIGU

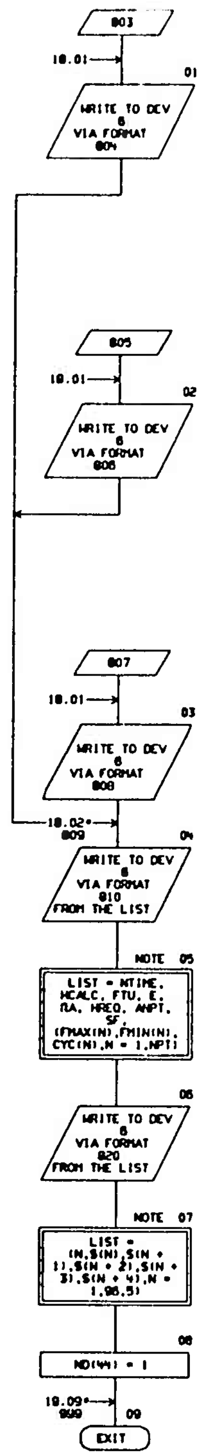


CHART TITLE - NON-PROCEDURAL STATEMENTS

```
      COMMON TCOM(5000)
      COMMON /IPRINT/ IP(80)
      DIMENSION D(2400), T(3000), ND(200)
      , CON(30), S(100), FMAX(300), FMIN(300), AN(300), CYC(300)
      , DAM1(300), DAM2(300)
      , PC1(300), PC2(300)
      , C1X(10), C1Y(10), C2X(10), C2Y(10)
      EQUIVALENCE (D(1),TCOM(1)), (T(1),TCOM(2401)), (ND(1),TCOM(5401))
      , (D(51),CON(1)), (D(101),H), (D(102),SF), (D(81),BLOCK)
      , (D(104),FTU), (D(105),E), (D(106),RA), (D(107),ANPT)
      , (D(201),FMAX(1)), (D(501),FMIN(1)), (D(801),AN(1))
      , (T(1),S(1)), (T(101),CYC(1)), (T(99),HREQ), (T(100),HCALC)
      , (T(401),PC1(1)), (T(701),PC2(1)),                 (S(42),RSF)
      , (T(1401),DAM1(1)), (T(1701),DAM2(1))
      , (T(1001),C1X(1)), (T(1011),C1Y(1)), (T(1021),C2X(1))
      , (T(1031),C2Y(1))
      , (CON(15),RTO), (CON(16),SGF), (CON(17),RTO1), (CON(18),RTO2)
      , (CON(19),C1), (CON(20),C2), (CON(21),BETA), (CON(22),GAMMA)
      , (CON(23),Q), (CON(24),RE), (CON(25),P), (CON(26),C1E)
      , (CON(27),UP1)
      EQUIVALENCE
       (ND(40),NPT) ,               (ND(42),NTOP)
      , (ND(43),NTIME)
      , (ND(45),M1), (ND(46),MTIME), (ND(47),K)
      , (S(87),SFH)
      , (S(88),SUMF), (S(89),FTOP)
      , (S(90),H1), (S(91),SF1), (S(92),H2), (S(93),SF2), (S(94),HN)
      , (S(95),SFN), (S(96),HR), (S(97),FACT), (S(98),TF)
15    FORMAT(1H1,88X,21H** FATIGU - IP(58) **//(I17,5E18.7))
40     FORMAT(13H0CURVE SET-UP // (4E20.7/ ))
41     FORMAT( 40X,2E20.7)
47     FORMAT( //(4F20.6 / ))
48     FORMAT( 40X,2F20.6 )
250   FORMAT(1H1,10X,43HNUMBER OF ITERATIONS IN SUBROUTINE FATIGU =,I4,
         31X,21H** FATIGU - IP(57) **/
                 /17X, 14HCALC. LIFE(HR), 9X 17HREQUIRED LIFE(HR), /
       16X,1E14.6,10X,1E14.6,6X,13HHIGHEST FMAX=,1E13.6,2H =,1F7.2,
       7HPCT FTU)
255   FORMAT(//8X,1HN, 7X, 4HFMAX, 8X, 4HFMIN, 5X, 8HAPP.CYC., 7X,
       8HDAMAGE 1,7X,3HPCT, 7X, 8HDAMAGE 2, 7X,3HPCT// (I19, 3F12.0,
       1E17.6, 1F8.2, 1E17.6, 1F8.2) )
257   FORMAT(// 13X,32HCUMULATIVE DAMAGE FOR EACH BLOCK, 2(E17.6,F8.2)
        //6X, 32HNUMBER OF RESIDUAL DAMAGE BLOCKS,1F9.3)
5012  FORMAT(1H1,88X,21H** FATIGU - IP(58) **)
330   FORMAT(13H0***** NTIME=,I13,7X,6HHCALC=,1E15.7,7X,7HFACTOR=,
       1E15.7,7X,3HTF=,1E15.7)
331   FORMAT (1H , 4E18.7)
505   FORMAT( 34H0 NEGATIVE LIFE THRU 5 ITTERATIONS )
802   FORMAT( 36H0WORKING ON WING AT SIDE OF FUSELAGE )
804   FORMAT( 36H0WORKING ON WING OUTER PANEL STATION )
806   FORMAT( 36H0WORKING OF FUSELAGE PRESSURE CYCLES )
808   FORMAT( 36H0WORKING ON FUSELAGE ENDURANCE LIMIT )
810   FORMAT(48H0*** ERROR IN FATIGUE, PERTINENT DATA FOLLOWS... /
       7H NTIME=,I13,4X,6HHCALC=,  1E14.6,4X,4HFTU=,1E14.6,4X,2HE=,
       1E14.6,4X,3HRA=,1E14.6 /14X,6HHREQ =,1E14.6,4X,5HANPT=,1E14.6,
       4X,3HSF=,1E14.6//15X,4HFMAX,14X,4HFMIN,14X,3HCYC /(6X,3E18.6) )
820   FORMAT(1H0,16HCALCULATION AREA // (I16, 5E18.7 ))
```

CHART TITLE - INTRODUCTORY COMMENTS

```
))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))
                     SUBROUTINE FTGCTL
((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((
```

CHART TITLE - SUBROUTINE FTGCTL

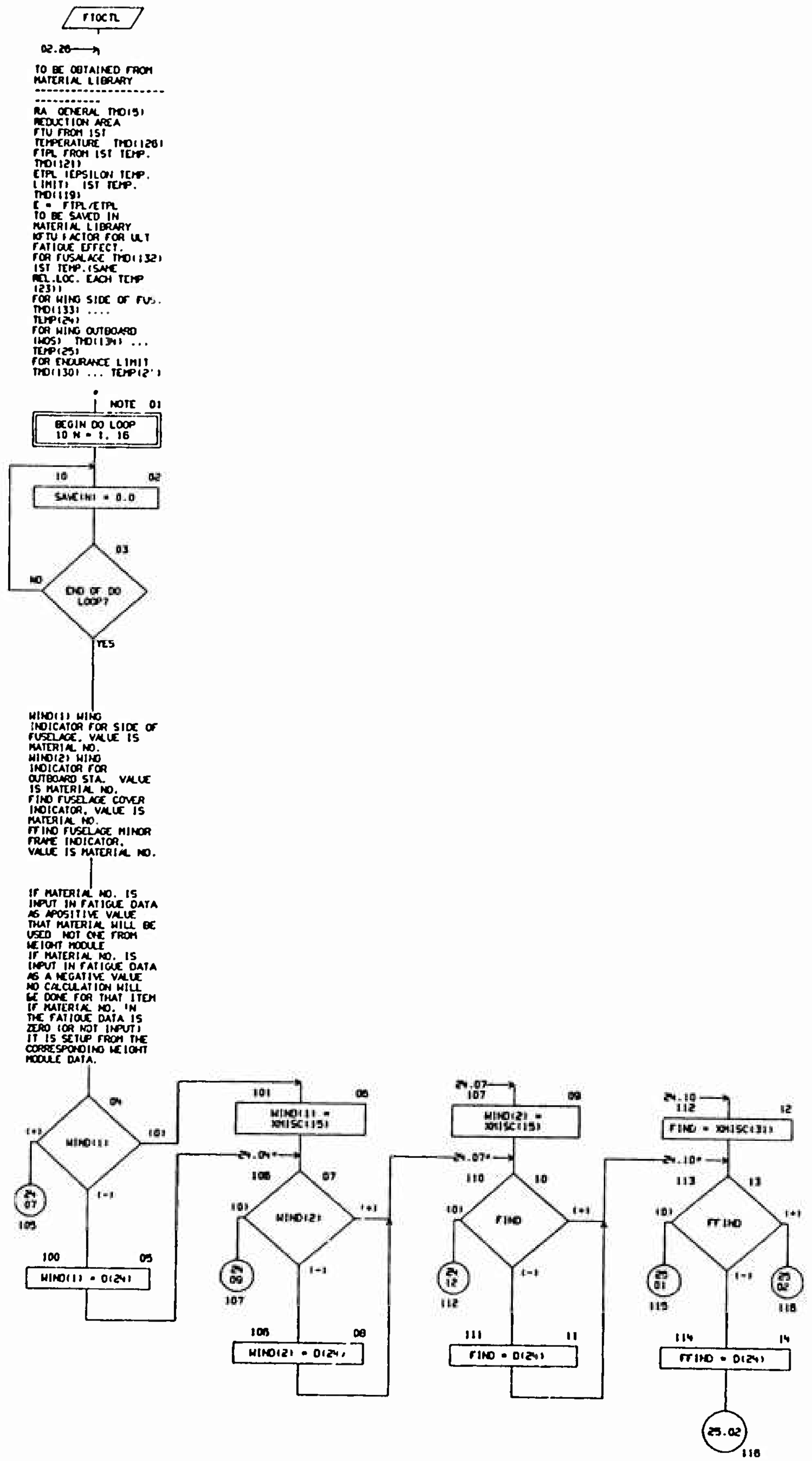


CHART TITLE - SUBROUTINE FTOCIL

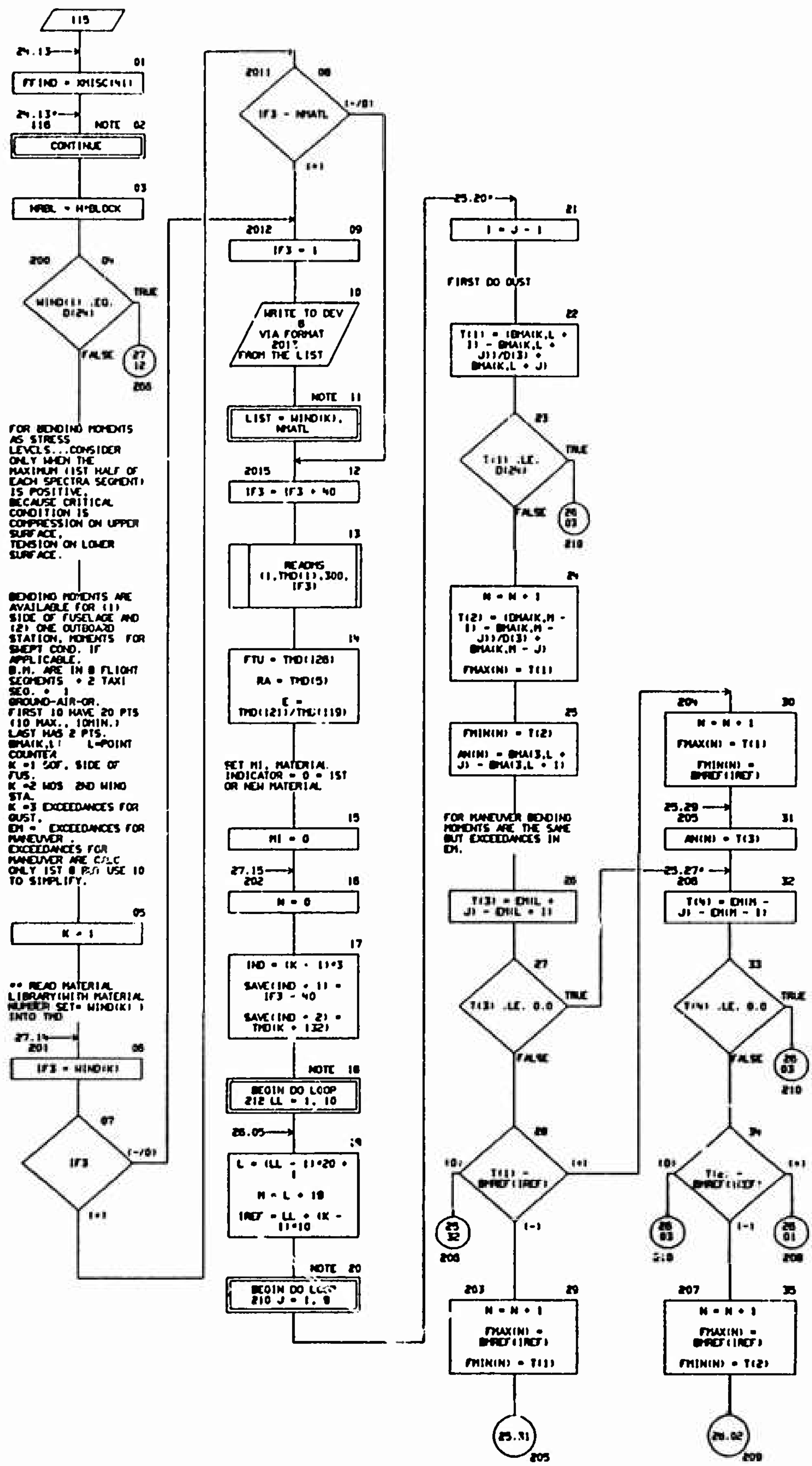


CHART TITLE - SUBROUTINE FTGCTL

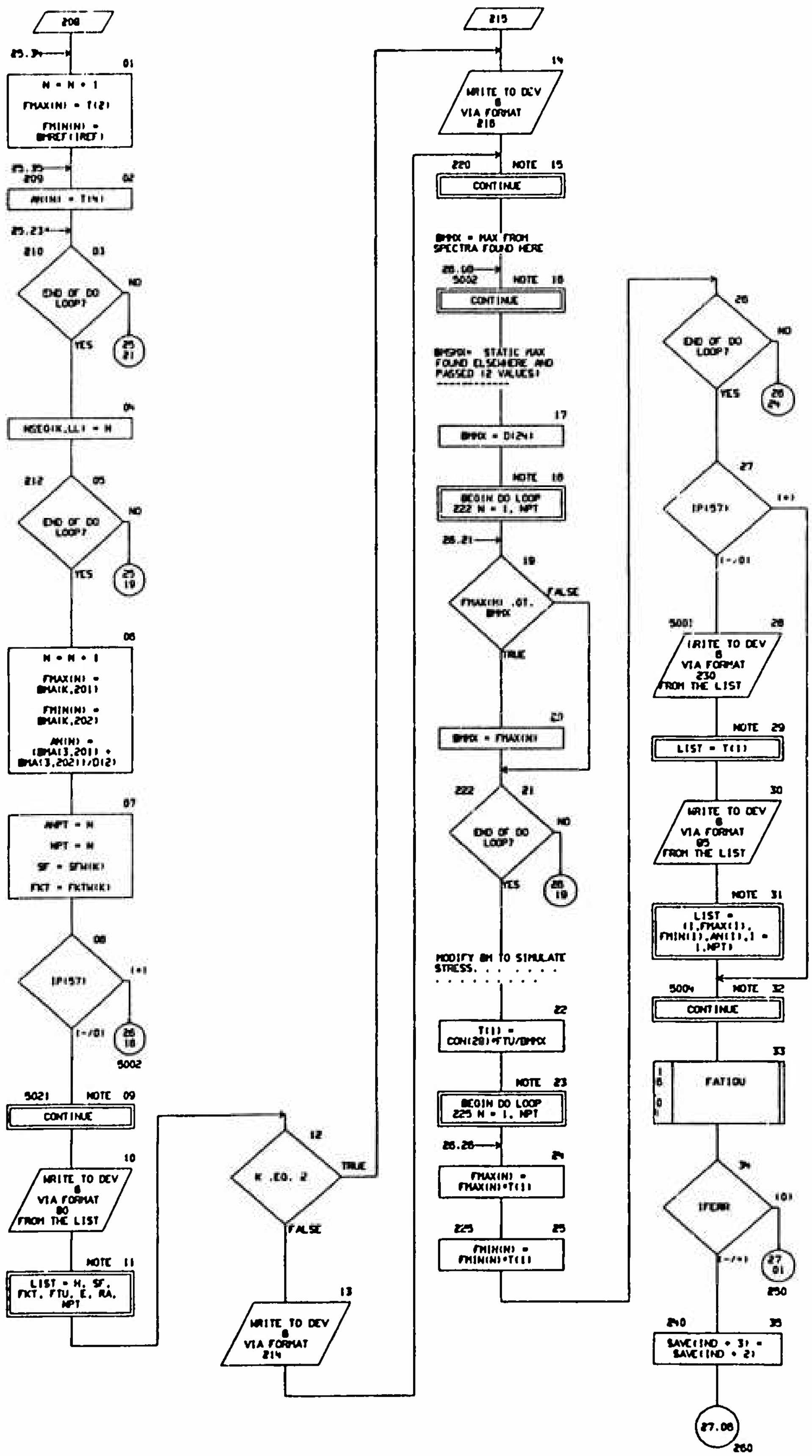
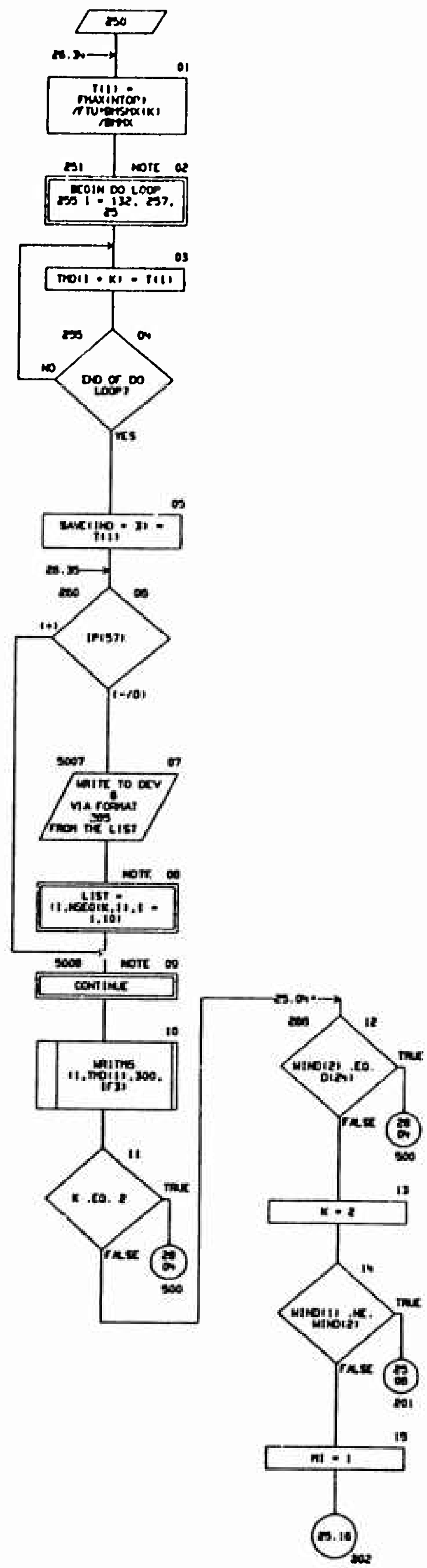


CHART TITLE - SUBROUTINE FTGCTL



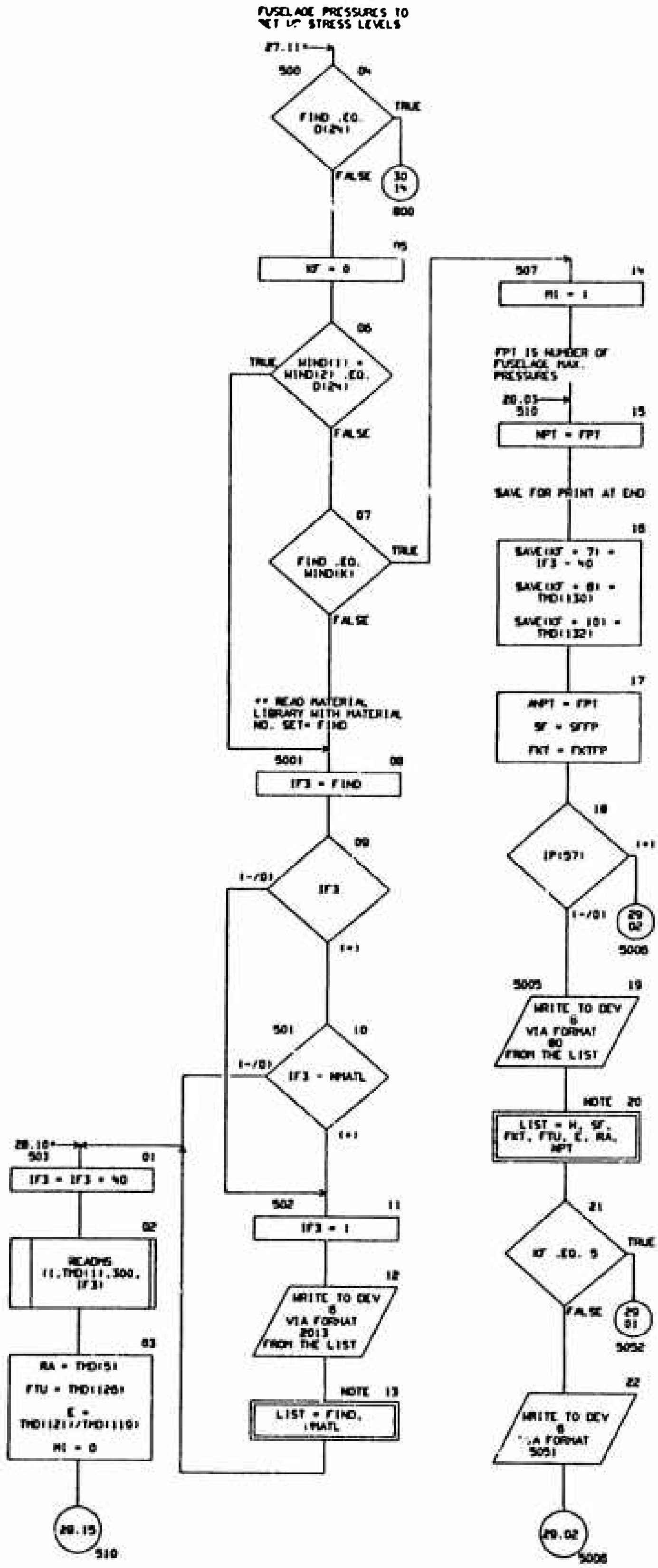


CHART TITLE - SUBROUTINE FTOCTL

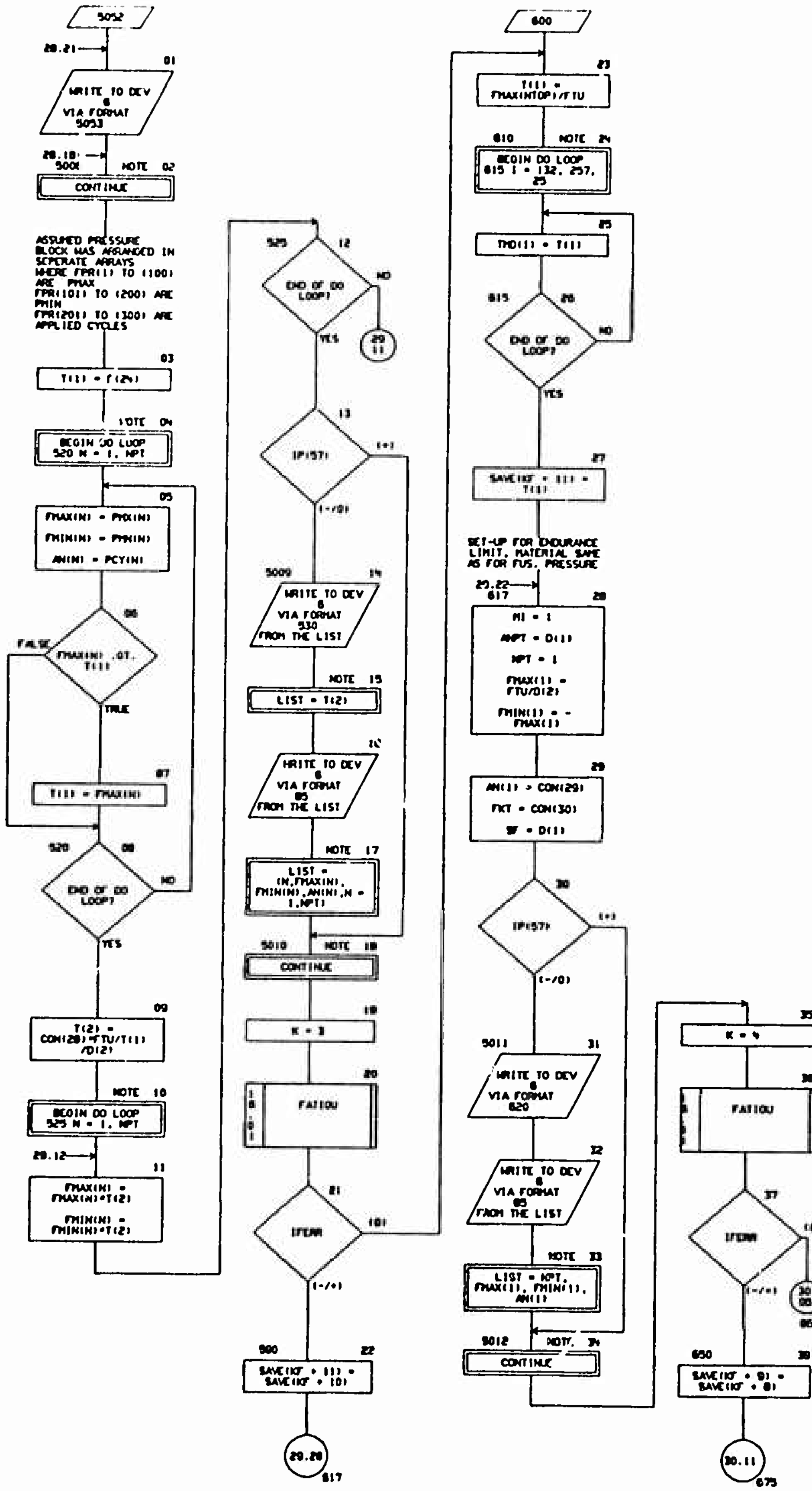


CHART TITLE - SUBROUTINE FIOCTL

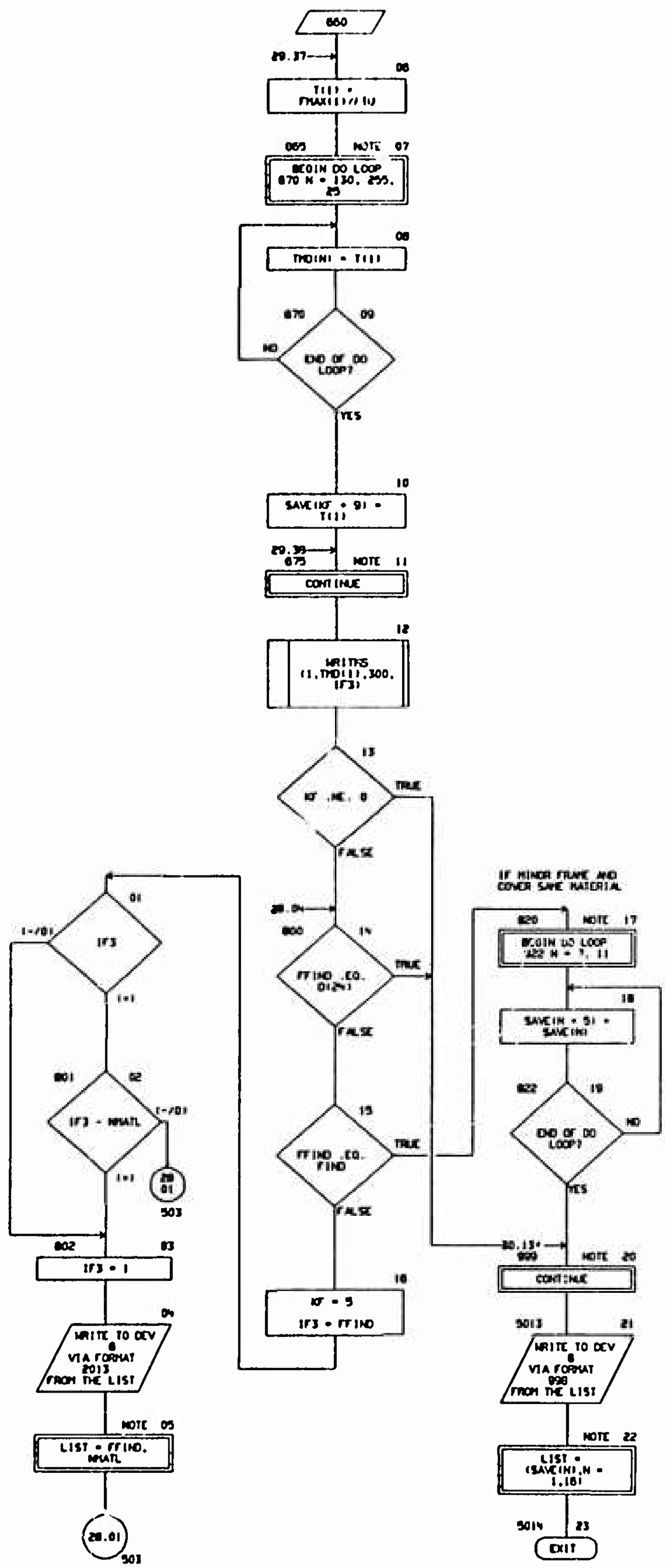


CHART TITLE - NON-PROCEDURAL STATEMENTS

```
      COMMON TCOM(5600)
      COMMON /MISC/XMISC(100)
      COMMON /IPRINT/ IP(80)
      DIMENSION SAVE(18)
      DIMENSION D(2400), T(3000), ND(200)
     , FMAX(300), FMIN(300), AN(300)
     , BMA(3,220), EM(220), BMREF(20), NSEQ(2,10)
     , BMSMX(2), THD(300)
     , MIND(2), SFM(2), FXTH(2)
     , CON(30)
     , FPR(300), PMX(100), PMN(100), PCY(100)
      EQUIVALENCE (D(1),TCOM(1)), (T(1),TCOM(2401)), (ND(1),TCOM(5401))
     .           (D(101),H), (D(102),SF), (D(103),FKT), (D(104),FTU)
     , (D(105),E), (D(106),RA), (D(107),ANPT)
     , (D(51),CON(1)), (D(81),BLOCK)
     , (D(109),MIND(1)), (D(111),FIND), (D(112),FFIND)
     , (D(113),SFM(1)), (D(115),FXTH(1)), (D(117),SFFP), (D(118),FKTFP)
     , (D(201),FMAX(1)), (D(501),FMIN(1)), (D(801),AN(1))
     , (D(1161),BMMX), (D(1162),BMSMX(1)), (T(1101),THD(1))
     , (D(1200),FPT), (D(1201),FPR(1)),         (T(41),NRBL)
     , (FPR(1),PMX(1)), (FPR(101),PMN(1)), (FPR(201),PCY(1))
     , (D(1501),BMA(1,1)), (D(2127),EM(1)), (D(2107),BMREF(1))
     , (ND(40),NPT)
     ,  (ND(121),NSEQ(1,1)), (ND(42),NTOP), (ND(44),IFERR)
     , (ND(45),MI), (ND(47),K)
     , (ND(93),IF3), (ND(59),NMATL)
2013  FORMAT(///1H0, 10H***** MATERIAL NO., F[illegible].0, 7H BEYOND, I14,
      18H, USED MATL. NO. 1  )
80    FORMAT(1H1,20HFATIGUE  INPUT DATA ,68X,21H** FTOCTL - IP(57) **/
                                         /1X, 2HH=,1E14.7, 5X, 3HSF=,
      1E14.7, 5X, 3HKT=,1E14.7//1X,4HFTU=,1E14.7, 5X, 2HE=,1E14.7, 5X,
      3HRA=, 1E14.7, 20X, 4HNPT=,I14  ////)
214   FORMAT(1[illegible]HOSIDE OF FUSELAGE )
218   FORMAT(22HOWING OUTBOARD STATION )
230   FORMAT( 42HOSTRESS LEVELS SET UP FROM BENDING MOMENTS,
     , 2X, 5HTIMES, 1E13.6 ////)
85    FORMAT(1I10, 3E19.7)
265   FORMAT(1H1,10X,41HN FOR END OF EACH SPECTRA SEGMENT FOLLOWS ,
      37X,21H** FTOCTL - IP(57) **// (11X,I13,I10))
5051  FORMAT(/ 15HOFUSELAGE COVER )
5053  FORMAT(/ 22HOFUSELAGE MINOR FRAMES)
530   FORMAT(52HOSTRESS LEVELS SET UP FROM FUSELAGE PRESSURES TIMES ,
       1E15.6 /// )
820   FORMAT(/// 18HOENDURANCE LIMIT  )
998   FORMAT(1H1,54HCHANGES MADE TO MATERIAL PROPERTIES BY FATIGUE PROGR
      AM,43X,12H** FTGCTL **/
        / 15X,27HSIDE OF FUSELAGE ** MATL NO,F4.0/
       11X,21HTHD(133) CHANGED FROM,F8.4,4H TO ,F8.4//
         16X,25HWING STATION 2 ** MATL NO,F4.0/
       11X,21HTHD(134) CHANGED FROM,F8.4,4H TO ,F8.4//
         18X,25 FUSELAGE COVER ** MATL NO,F4.0/
       11X,21HTHD(130) CHANGED FROM,F8.4,4H TO ,F8.4/
       11X,21HTHD(132) CHANGED FROM,F8.4,4H TO ,F8.4//
         13X,31HFUSELAGE MINOR FRAME ** MATL NO,F4.0/
       11X,21HTHD(130) CHANGED FROM,F8.4,4H TO ,F8.4/
       11X,21HTHD(132) CHANGED FROM,F8.4,4H TO ,F8.4)
```

FORTRAN LISTING

OF

FATIGUE MODULE

FORTRAN MODULE (LIST AUTOSEQ)

CARD NO **** CONTENTS ****

```
      PROGRAM FATGUE
C
C
      COMMON TCOM(5600)                                                 90000015
      COMMON /MISC/XMISC(100)
      COMMON /IPRINT/IP(80)
C
      DIMENSION D(2400), T(3000), ND(200)                               90000020
     1, BMA(3,220), DUMMY(830), EM(220), BMREF(20), BMSMX(2)            90000021
C
      EQUIVALENCE (D(1),TCOM(1)), (T(1),TCOM(2401)), (ND(1),TCOM(5401)) 90000030
     1, (BMA(1,1),DUMMY(1),D(1501)), (BMREF(1),D(2107)), (EM(1),D(2127))90000031
     2, (BMSMX(1),D(1162))                                              90000032
     *, (ND(59),NMATL)                                                  90000050
C                                                                       90000080
      DO 10 I=1,5600                                                    90000090
   10 TCOM(I) = 0.0                                                     90000100
C
      CALL READMS(1,D(1),2400,29)
C
      IF(D(1162))20,20,22
   20 BMSMX(1) = XMISC(32)
   22 IF(D(1163))23,23,24
   23 BMSMX(2) = XMISC(33)
   24   IF (D(101)) 25,25,26
   25 D(101) = XMISC(34)
   26   CONTINUE
C         SO THE WIND WILL HAVE THE SAME VALUE
      XMISC(32)=BMSMX(1)
      XMISC(33)=BMSMX(2)
C                                                                       10000130
      NMATL = XMISC(1)
C
      CALL READMS(1,DUMMY(1),830,35)
C
C
      IF(IP(56))5001,5001,5002
 5001 CONTINUE
      WRITE(6,140)
  140 FORMAT(1H1,70H*** BMA(3,220) FROM SUBROUTINE FATMD IN LOADS PROGRA
     *M IN RECORD 35 ***,10X,21H** FATGUE - IP(56) **//
     *  8X, 12HSOF BEND MOM, 3X, 12HWOS BEND MOM, 2X,
     *  16HEXCEEDANCES-GUST, 2X, 16HEXCEEDANCES-MANU  )
C
  150  WRITE(6,151) (N,BMA(1,N),BMA(2,N),BMA(3,N),EM(N),N=1,202)
  151  FORMAT( I15,2F15.0,2E18. 5)
      WRITE(6,153) (N,DUMMY(N+606), DUMMY(N+818),N=1,8)                 9000
  153  FORMAT( 39HOREFERENCE BENDING MOMENTS FOR MANEUVER / 4X, 7HSEGMENT9000
     1 8X, 3HSOF, 10X, 3HWOS /(I10, 2F15.0 ) )                          9000
C
      WRITE(6,152)BMSMX(1),BMSMX(2)
  152 FORMAT(//10X,10HBMSMX(1) =,F15.0/10X,10HBMSMX(2) =,F15.0//)
 5002 CONTINUE
C
      CALL FTOCTL                                                       90000320
C
      END                                                               90000490
C
C )))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))
C                          FUNCTION ACURVE
C ((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((
C
      FUNCTION ACURVE(X)                                                90040010
C                          FOR CYCLIC STRESS, CYCLIC STRAIN CURVE       90040020
      COMMON TCOM(5600)                                                 90040030
C                                                                       90040040
      DIMENSION D(2400), T(3000), ND(200)                               90040050
     *, CON(50), S(100)                                                 90040060
     *, CIX(10), CIY(10)                                                90040080
C                                                                       90040090
```

CARD NO **** CONTENTS ****

```
      EQUIVALENCE (D(1),TCOM(1)), (T(1),TCOM(2401)), (ND(1),TCOM(5401)) 90040100
     *, (D(51),CON(1)), (D(105),E)                                      90040110
     *, (T(1),S(1))                                                     90040140
     *, (T(1001),CIX(1)), (T(1011),CIY(1))                              90040170
     *,             (CON(20),C2), (CON(23),Q), (CON(26),CIE)            90040200
C              NOTE SCRATCH AREA FOR ROUTINE IS S(20) THRU (28)         90040240
      EQUIVALENCE (S(20),XI), (S(21),XCL), (S(22),XJ), (S(23),YJ)       90040250
     *, (S(24),XK), (S(25),YK), (S(26),YI), (S(27),YREAL), (S(29),XCR)  90040260
      DATA XPRIOR/0.0/, YPRIOR/0.0/                                     90040270
      IFUP=0                                                            90040280
      IF(ABS(X-XPRIOR)/X .LE. 0.000001 )   GO TO 999                    90040290
      XPRIOR = X                                                        90040300
      K=1                                                               90040310
      XI = ALOG(X)                                                      90040320
      IF(XI - CIX(1)) 10,10,15                                          90040330
C                         FOR NOW IF LESS THAN 1ST CURVE PT. SAY = TO.  90040340
   10 YPRIOR = EXP(CIY(K))                                              90040350
      GO TO 999                                                         90040360
   15 K=9                                                               90040370
      IF(CIX(9) - XI) 20,10,30                                          90040380
   20 IFUP=1                                                            90040390
      GO TO 40                                                          90040400
   30 DO 35 N=2,9                                                       90040410
         K=N                                                            90040420
         IF(XI-CIX(K)) 40,10,35                                         90040430
   35 CONTINUE                                                          90040440
   40 YK = CIY(K)                                                       90040450
      XK = CIX(K)                                                       90040460
      YJ = CIY(K-1)                                                     90040470
      XJ = CIX(K-1)                                                     90040480
   45 YI = (YK-YJ)/(XK-XJ) * (XI-XJ) + YJ                               90040490
      YREAL = EXP(YI)                                                   90040500
      XCR = YREAL/E + C2*(YREAL/CIE)**Q                                 90040510
      XCL = ALOG(XCR)                                                   90040520
C                                                                       90040530
C                                                                       90040540
      IF ( ABS(XCR-X)/X - CON(12)) 998,998,50                           90040550
   50 IF(IFUP) 55,55,70                                                 90040560
   55 IF(XCL-XI) 60,998,65                                              90040570
   60 XJ = XCL                                                          90040580
      YJ = YI                                                           90040590
      GO TO 45                                                          90040600
   65 XK = XCL                                                          90040610
      YK = YI                                                           90040620
      GO TO 45                                                          90040630
   70 IF(XCL-XI) 75,998,72                                              90040640
   72 IFUP = 0                                                          90040650
   75 XJ = XK                                                           90040660
      YJ = YK                                                           90040670
      XK = XCL                                                          90040680
      YK = YI                                                           90040690
      GO TO 45                                                          90040700
C                                                                       90040710
  998 YPRIOR = YREAL                                                    90040720
  999 ACURVE = YPRIOR                                                   90040730
      RETURN                                                            90040740
      END                                                               90040750
C
C )))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))
C                              SUBROUTINE ALIFE
C (((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((
C
      SUBROUTINE ALIFE                                                  90030010
      COMMON TCOM(5600)                                                 9003
C              MAY 1972                                                 90030030
C                                                                       90030040
      DIMENSION D(2400), T(3000), ND(200)                               9003
     1, CON(30), S(100), FMAX(300), FMIN(300),          CYC(300)        90030060
     2, DEPS(300), SOMEN(300), SCR1(300), DAM1(300), DAM2(300)          9003
C                                                                       90030090
      EQUIVALENCE (D(1),TCOM(1)), (T(1),TCOM(2401)), (ND(1),TCOM(5401)) 9003
```

```
CARD NO   ****                        CONTENTS                                   ****

          *, (D(51),CON(1)),                          (D(103),FKT)               90030110
          *, (D(104),FTU), (D(105),E)                                            90030120
          *, (D(201),FMAX(1)), (D(501),FMIN(1))                                  90030130
          *, (T(1),S(1)), (T(101),CYC(1)), (T(99),HREQ), (T(100),HCALC)          90030140
          *, (T(1401),DAM1(1)), (T(1701),DAM2(1))                                9003
          *, (T(2001),DEPS(1)), (T(2301),SGR1(1)), (T(2601),SGMEN(1))            9003
          *,                  (CON(16),SO'), (CON(20),C2)                        90030190
          *, (CON(23),Q),                    (CON(25),P), (CON(26),C1E)          90030210
          *, (CON(27),QP1)                                                       90030220
          *,(IND(40),NPT), (IND(41),JMAX)                                        90030230
           EQUIVALENCE (S(41),HRDL), (S(42),RSF)                                 90030235
           EQUIVALENCE (S(51),A), (S(52),B), (S(53),F), (S(54),X), (S(55),FA)90030240
          *, (S(58),F1), (S(57),DF2),                (S(59),FSQ)                 90030250
          *,                                 (S(63),DSO1), (S(64),DSO2)          90030260
          *, (S(65),EPSA), (S(66),SOMN), (S(67),SOMX)                            90030270
          *, (S(69),TNF1), (S(70),TNF2), (S(71),DEPS1), (S(72),DEPS2)            90030280
          *, (S(73),DSOR1), (S(74),DSOR2), (S(75),EPSOT), (S(76),EPSMX)          90030290
          *, (S(77),EPSTJ), (S(78),EPST2), (S(79),SDAM1), (S(80),SDAM2)          90030300
          *,                  (S(82),DEPSMX), (S(83),DEPSR2), (S(84),SOBMEN)     90030310
          *, (S(85),SORIMX), (S(86),TEPSTJ)                                      90030320
      C                                                                          90030330
           HCALC = D(24)                                                         90030340
           DEPSMX = D(24)                                                        90030350
           SDAM1 = D(24)                                                         90030360
           SDAM2 = D(24)                                                         90030370
      C                                                                          90030380
      C      NPT = NO. OF FMAX-FMIN-CYC VALUES        MAX=300.                   90030390
      C                                                                          90030400
      C ......................................................................90030410
      C                                                                          90030420
           DO 250 J=1,NPT                                                        90030430
           FSQ=((FKT*FMAX(J))**2)/CON(10)                                        90030440
           A = FTU/D(14)                                                         90030450
           B = D(24)                                                             90030460
           I=0                                                                   90030470
       90  X=A                                                                   90030480
           FA = A**2 + P*A**QP1 - FSQ                                            90030490
           IF(FA)95,170,97                                                       90030500
       95  I=I+1                                                                 90030510
           IF(I .GT. 5) GO TO 166                                                90030520
           A=A*C(10)                                                             90030530
           GO TO 90                                                              90030540
       97  I=0                                                                   90030550
      100  IF(I .GT.  40 ) GO TO 166                                             90030560
           I=I+1                                                                 90030570
           X = (A+B)/D(2)                                                        90030580
           F = X**2 + P*X**QP1 - FSQ                                             90030590
           IF(F) 110,170,120                                                     90030600
      110  IF(F+CON(1)) 130,170,170                                              90030610
      120  IF(F-CON(1)) 170,170,130                                              90030620
      130  IF(F*FA) 140,170,160                                                  90030630
      140  IF(B-X) 150,170,170                                                   90030640
      150  B = X                                                                 90030650
           GO TO 100                                                             90030660
      160  A = X                                                                 90030670
           FA = F                                                                90030680
           GO TO 100                                                             90030690
      C      CANT FIND ROOT - - HCALC WILL BE ZERO ON RETURN                     90030700
      166  WRITE(6,167) I,X,F,A,FA,B                                             90030710
      167   FORMAT(1H1,47HFAILED TO FIND SOMX IN ALIFE SUBROUTINE ******* // 90030720
          * 3H I=,I13, 2X,2HX=,1E14.6,2X,2HF=,1E14.6,2X,2HA=,1E14.6,2X,3HFA=,90030730
          * 1E14.6,2X,2HB=,1E14.6 )                                              90030740
           GO TO 999                                                             90030750
      C                                                                          90030760
      170  SOMX = X*D(14)                                                        90030770
           EPSMX=SOMX/E + C2*(SOMX/C1E)**Q                                       90030780
      C                                                                          90030790
           IF(FMIN(J) .GE. D(24)) F1=(FMAX(J)+FMIN(J))/D(2)                      90030800
           IF(FMIN(J) .LT. D(24)) F1=FMAX(J)/D(2)                                90030810
           DF2 = F1-FMIN(J)                                                      90030820
           DSO1 = FKT*(FMAX(J)-F1)                                               90030830
```

```
CARD NO    ****                         CONTENTS                              ****

                 UC"S1= DSO1/E                                              90030840
                 DEPS2= DF2 * EPSMX/ FMAX(J)                                90030850
                 DEPS(J)= DEPS1+DEPS2                                       90030860
                 EPSTJ = ABS(DEPS2)                                         90030870
                 DSO2 = ACURVE(EPSTJ)                                       90030880
                 IF(DEPS2 .LT. D(241)) DSO2=-DSG2                           90030890
                 SOMN = SGMX-DSO1-DSO2                                      90030900
                 SOMEN(J) = (SGMX+SGMN)/D(2)                                90030910
                 EPSA = DEPS(J)/D(2)                                        90030920
                 EPST2 = EPSA/(D(1)-(SGMEN(J)/SOF))                         90030930
                 TEPSTJ= ABS(EPST2)                                         90030940
                 TNF1 = BCURVE(TEPSTJ)                                      90030950
                 DAM1(J) = CYC(J)/TNF1                                      90030960
           C     DAMAGE DUE TO JTH STRESS  FROM FIRST                       90030970
           C                                                                90030980
                 SDAM1 = SDAM1 + DAM1(J)                                    90030990
           C                                                                90031000
                 IF(FMIN(J) .GE. 0.0 .AND. FMAX(J) .GE. 0.0 ) GO TO 200     90031010
                 IF(FMI.(J) .LT. 0.0 .AND. FMAX(J) .LT. 0.0 ) GO TO 200     90031020
                 DSGR1 = FKT*FMIN(J)/D(2)                                   90031030
                 DEPSR2=  FMIN(J)/D(2)*EPSMX/FMAX(J)                        90031040
                 S(1) = SGMN                                                90031050
                 GO TO 210                                                  90031060
             200 DSGR1 = FKT*FMAX(J)/D(2)                                   90031070
                 DEPSR2= EPSMX/D(2)                                         90031080
                 S(1) = SGMX                                                90031090
             210 EPSTJ = ABS(DEPSR2)                                        90031100
                 DSGR2 = ACURVE(EPSTJ)                                      90031110
                 IF(DEPSR2 .LT. D(241)) DSGR2 = -DSGR2                      90031120
                 SGR1(J) = S(1)-DSGR1-DSGR2                                 90031130
           C                                                                90031140
           C                                                                90031150
                 IF(DEPS(J) .LE. DEPSMX ) GO TO 250                         90031320
                        DEPSMX = DEPS(J)                                    90031330
                        JMAX = J                                            90031340
             250 CONTINUE                                                   90031350
           C                                                                90031360
                 SGR1MX = SGR1(JMAX)                                        90031370
           C                                                                90031380
           C ...............................................................90031390
           C                                                                90031400
                 DO 300 J=1,NPT                                             90031410
                 SGBMEN = SGMEN(J) + SGR1MX - SGR1(J)                       90031420
                 EPSBT = DEPS(J)/D(2) / (D(1)-SGBMEN/SOF)                   90031430
                 TEPSTJ = ABS(EPSBT)                                        90031440
                 TNF2 = BCURVE(TEPSTJ)                                      90031450
                 DAM2(J) = CYC(J)/TNF2                                      90031460
                 SDAM2 = SDAM2 + DAM2(J)                                    90031470
           C                                                                ?0031480
           C                                                                90031530
             300 CONTINUE                                                   90031540
           C                                                                90031550
           C                                                                90031600
                 TCALC = ((( RSF-SDAM1)/SDAM2) + D(1)) * HRBL               90031610
           C                                                                90031620
             999 RETURN                                                     90031630
                 END                                                        90031640
           C
           C )))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))
           C                           FUNCTION BCURVE
           C (((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((
           C
                 FUNCTION BCURVE(X)                                         90050010
           C     TO GET Y FOR THE  STRAIN VS CYCLES TO FAILURE CURVE        90050020
           C           X =  C1/ Y**BETA + C2/ Y**GAMMA                      90050030
           C                                                                90050040
                 COMMON TCOM(5600)                                          ?0050050
           C                                                                90050060
                 DIMENSION J(2400), T(3000), ND(200)                        90050070
                *, CON(50), S(100)                                          90050080
                *,                     C2X(10), C2Y(10)                     90050100
```

CARD NO **** CONTENTS ****

```
      C                                                                 90050110
            EQUIVALENCE (D(1),TCOM(1)), (T(1),TCOM(2401)), (ND(1),TCOM(5401)) 90050120
           *, (D(51),CON(1))                                            90050130
           *, (T(1),S(1))                                               90050160
           *, (T(1021),C2X(1)), (T(1031),C2Y(1))                        90050200
           *, (CON(19),C1), (CON(20),C2), (CON(21),BETA), (CON(22),GAMMA)  90050220
      C              NOTE SCRATCH FOR ROUTINE IS S(30) THRU (38)        90050260
            EQUIVALENCE (S(30),XI), (S(31),XCL), (S(32),XJ), (S(33),YJ) 90050270
           *, (S(34),XK), (S(35),YK), (S(36),YI), (S(37),YREAL), (S(38),XCR)  90050280
      C                                                                 90050290
      C ............................. ................................. 90050300
      C                                                                 90050310
            XI = ALOG(X)                                                90050320
            K=1                                                         90050330
            IF(XI-C2X(1)) 20,10,10                                      90050340
         10 YREAL = EXP(C2Y(K))                                         90050350
            GO TO 999                                                   90050360
         20 K=10                                                        90050370
            IF(XI-C2X(10)) 10,10,30                                     90050380
      C                          START WITH NO EXTRAPOLATION            90050390
         30 DO 35 N=2,10                                                90050400
                K=N                                                     90050410
                IF(C2X(N)-XI) 40,10,35                                  90050420
         35 CONTINUE                                                    90050430
        40  XK=C2X(K)                                                   90050440
            YK=C2Y(K)                                                   90050450
            XJ=C2X(K-1)                                                 90050460
            YJ=C2Y(K-1)                                                 90050470
      C                                                                 90050480
         45 YI=(YK-YJ)/(XK-XJ)*(XI-XJ) + YJ                             90050490
            YREAL = EXP(YI)                                             90050500
            XCR = C1/YREAL**BETA + C2/YREAL**GAMMA                      90050510
            XCL = ALOG(XCR)                                             90050520
      C                                                                 90050530
      C                                                                 90050540
            IF( ABS(XCR-X)/X - CON(12)) 999,999,50                      90050550
         50 IF(XCL-XI) 60,999,55                                        90050560
         55 XJ = XCL                                                    90050570
            YJ = YI                                                     90050580
            GO TO 45                                                    90050590
         60 XK = XCL                                                    90050600
            YK = YI                                                     90050610
            GO TO 45                                                    90050620
        999 BCURVE = YREAL                                              90050630
            RETURN                                                      90050640
            END                                                         90050650
      C
      C )))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))
      C                       SUBROUTINE FATIGU
      C (((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((
      C
            SUBROUTINE FATIGU                                           90020010
      C     .........................................................   90020020
      C                                                                 90020030
      C *** FATIGUE ROUTINE CALCULATES LIFE FROM ARRAY OF STRESS MAX + MIN   90020040
      C        AND APLICABLE CYCLES PER REQUIRED HOURS.                 90020050
      C                                                                 90020060
      C     MATERIAL PROPERTIES USED ARE      FTU - TENSILE STRENGTH    90020070
      C                                       E   - MODULUS OF ELASTICITY   90020080
      C                                       RA  - REDUCTION IN AREA   90020090
            COMMON TCOM(5600)                                           90020100
            COMMON /IPRINT/ IP(80)
      C                                                                 90020110
            DIMENSION D(2400), T(3000), ND(200)                         90020120
           1, CON(30), S(100), FMAX(300), FMIN(300), AN(300), CYC(300)  90020130
           2, DAM1(300), DAM2(300)                                      90020140
           3, PC1(300), PC2(300)                                        90020150
           *, C1X(10), C1Y(10), C2X(10), C2Y(10)                        90020160
      C                                                                 90020170
            EQUIVALENCE (D(1),TCOM(1)), (T(1),TCOM(2401)), (ND(1),TCOM(5401)) 90020180
           *, (D(51),CON(1)), (D(101),H), (D(102),SF), (D(81),BLOCK)    90020190
```

CARD NO **** CONTENTS ****

```
     *, (D(104),FTU), (D(105),E), (D(106),RA), (D(107),ANPT)                 90020200
     *, (D(201),FMAX(1)), (D(501),FMIN(1)), (D(801),AN(1))                    90020210
     *, (T(1),S(1)), (T(10),CYC(1)), (T(99),HREQ), (T(100),HCALC)             90020220
     *, (T(40),PC1(1)), (T(70),PC2(1)),                          (S(42),RSF)  90020230
     *, (T(140),DAM1(1)), (T(170),DAM2(1))                                    90020240
     *, (T(1001),C1X(1)), (T(101),C1Y(1)), (T(1021),C2X(1))                   90020260
     *, (T(1031),C2Y(1))                                                      90020270
     *, (CON(15),RTO), (CON(16),SOF), (CON(17),RTO1), (CON(18),RTO2)          90020200
     *, (CON(19),C1), (CON(20),C2), (CON(21),BETA), (CON(22),GAMMA)           90020290
     *, (CON(23),Q), (CON(24),RE), (CON(25),P), (CON(26),C1E)                 90020300
     *, (CON(27),QP1)                                                         90020310
      EQUIVALENCE                                                             90020320
     * (ND(40),NPT) ,                 (ND(42),NTOP)                           90020330
     *, (ND(43),NTIME)                                                        90020340
     *, (ND(45),MI), (ND(46),MTIME), (ND(47),K)                               90020350
     *, (S(87),SFM)                                                           90020370
     *, (S(88),SUMF), (S(89),FTOP)                                            90020380
     *, (S(90),H1), (S(91),SF1), (S(92),H2), (S(93),SF2), (S(94),HN)          90020390
     *, (S(95),SFN), (S(96),HR), (S(97),FACT), (S(98),TF)                     90020400
C                                                                             90020410
C ........................................ ................................. 90020420
C     INITALIZE                                                               90020430
C                                                                             90020440
      ND(44) = 0                                                              90020450
      MTIME = 0                                                               90020451
C     IF MATERIAL SAME SKIP COEFFICIENT AND CURVE CALCULATION                 90020460
      IF(MI .NE. 0) GO TO 49                                                  90020470
      RTO = ALOG(D(1)/ (D(1)-RA))                                             90020480
      SOF = FTU*(D(1)+RTO)                                                    90020490
      RTO1= FTU/E                                                             90020500
      RTO2= (D(1)+RTO)                                                        90020510
      C1 = CON(1)*RTO1*RTO2**CON(2)                                           90020520
      BETA = CON(3)+CON(4)*ALOG10(RTO2)                                       90020530
      S(1) = CON(5)* RTO**CON(6                                               90020540
      S(2) = (D(1)-(CON(7)*RTO1)*(RTO2**CON(4)))**D(2))                       90020550
      GAMMA = ALOG10(S(1)/S(2))                                               90020560
      C2 = (RTO**CON(8))/D(8)*(D(10)**GAMMA)                                  90020570
      Q = GAMMA/BETA                                                          90020580
      RE = E/D(14)                                                            90020590
      P = RE*C2/(RE*C1:**                                                     90020600
C                                                                             90020610
      C1E = C1*E                                                              90020620
      QP1 = D(1)+Q                                                            90020630
C                                                                             90020640
      IF(IP(58))5001,5001,17
 5001 CONTINUE
      WRITE (6,15) (N,D(N),D(N+1),D(N+2),D(N+3),D(N+4),N=65,79,5)             90020660
   15 FORMAT(1H1,88X,21H** FATIGU - IP(58) **//(I17,5E16.7))
 17   CONTINUE                                                                90020680
C                                                                             90020690
C ......................................................................... 90020700
C                                                                             90020710
C  SET UP CURVE     CYCLIC STRESS VS CYCLIC STRAIN --- ACURVE                 90020720
C         EPST = SOA/E + C2(SOA/C1E)**Q                                       90020730
C                 EPST = C1X                                                  90020740
C                 SOA  = C1Y                                                  90020750
C                                                                             90020760
C      AND STRAIN VS CYCLES TO FAILURE                                        90020770
C         TEPST = C1/TNF**BETA + C2/TNF**GAMMA    --- BCURVE                  90020780
C                 TEPST = C2X                                                 90020790
C                 TNF   = C2Y                                                 90020800
C                                                                             90020810
C  THESE CURVES ARE NEAR-LINEAR  ON LOG LOG PLOT SO WILL USE LOGS AND        90020820
C      STRAIGHT LINE INTERPOLATION IN ITERATING TO FIND Y FOR A GIVEN X.     90020830
C                                                                             90020840
C      FOR A  THERE IS A SLOPE CHANGE WHERE THE DOMINANT TERM CHANGES         90020850
C          Y DISTRIBUTION IS PICKED FOR THIS.                                 90020860
C                                                                             90020870
C ......................................................................... 90020880
C                                                                             90020890
      C1Y(1) = D(1)                                                           90020900
```

CARD NO **** CONTENTS ****

```
          C1Y(9) = CON(9)                                           90020910
          C1Y(5) = E*(C1**Q/C2)**(D(1)/(Q-D(1)))                    90020920
          C1Y(4) = C1Y(5)/D(2)                                      90020930
          C1Y(2) = C1Y(4)**D(2))                                    90020940
          C1Y(3) = C1Y(2)**2                                        90020950
          C1Y(6) = C1Y(5) + C1Y(4)                                  90020960
          S(1) = (C1Y(9)/C1Y(6))**D(2))                             90020970
          C1Y(7) = S(1)*C1Y(6)                                      90020980
         C1Y(8) = C1Y(7)*S(1)                                       90020990
          DO 30 N=1,9                                               90021000
              C1X(N) = C1Y(N)/E + C2*(C1Y(N)/C1E)**Q                90021010
     30   CONTINUE                                                  90021020
C        FOR B     NOTE- X IN DECREASING ORDER IN STORAGE    *****  90021030
          C2Y(1) = D(1)                                             90021040
         DO 32 N=2,6                                                90021050
         C2Y(N) = C2Y(N-1)*D(10)                                    90021060
    32   CONTINUE                                                   90021070
         S(1) = D(10)                                               90021080
C        AS CURVE FLATTENS INCREASE INTERVAL TO COVER RANGE BETTER. 90021090
         DO 35 N=7,10                                               90021100
         S(1) = S(1)*D(10)                                          90021110
         C2Y(N) = C2Y(N-1)*S(1)                                     90021120
    35   CONTINUE                                                   90021130
          DO 38 N=1,10                                              90021140
              C2X(N) = C1/C2Y(N)**BETA + C2/C2Y(N)**GAMMA           90021150
     38   CONTINUE                                                  90021160
C                                                                   90021170
         IF(IP(50))5003,5003,42
  5003 CONTINUE
          WRITE(6,40) (C1X(N),C1Y(N),C2X(N),C2Y(N), N=1,9)          90021190
     40   FORMAT(13HOCURVE SET-UP // (4E20.7/ ))                    90021200
          WRITE(6,41) C2X(10),C2Y(10)                               90021210
     41   FORMAT( 40X,2E20.7)                                       90021220
     42  CONTINUE                                                   90021230
C                                                                   90021240
          DO 45 N=1,9                                               90021250
          C1X(N) = ALOG(C1X(N))                                     90021260
          C1Y(N) = ALOG(C1Y(N))                                     90021270
          C2X(N) = ALOG(C2X(N))                                     90021280
          C2Y(N) = ALOG(C2Y(N))                                     90021290
     45   CONTINUE                                                  90021300
          C2X(10) = ALOG(C2X(10))                                   90021310
          C2Y(10) = ALOG(C2Y(10))                                   90021320
C                                                                   90021330
         IF(IP(50))5005,5005,49
  5005 CONTINUE
          WRITE(6,47) (C1X(N),C1Y(N),C2X(N),C2Y(N), N=1,9)          90021350
     47   FORMAT( //(4F20.6 / ))                                    90021360
          WRITE(6,48) C2X(10),C2Y(10)                               90021370
     48   FORMAT( 40X,2F20.6 )                                      90021380
     49  CONTINUE                                                   90021390
C                                                                   90021400
C ... ............................................................. 90021410
C                                                                   90021420
         NPT = ANPT                                                 90021430
         FTOP = D(24)                                               90021440
         HREQ = H*SF                                                90021450
         RSF=D(1)/SF                                                90021455
         SUMF = D(24)                                               90021460
         DO 100 N=1,NPT                                             90021470
         CYC(N) = AN(N)*BLOCK                                       90021480
         SUMF = SUMF + FMAX(N)                                      90021490
         IF(FMAX(N) .LE. FTOP ) GO TO 100                           90021500
                  FTOP = FMAX(N)                                    90021510
                  NTOP = N                                          90021520
   100 CONTINUE                                                     90021530
C                                                                   90021540
         DO 110 N=90,98                                             90021550
   110   S(N) = D(24)                                               90021560
         HR = ALOG(HREQ)                                            90021570
         TF = D(1)                                                  90021580
```

CARD NO **** CONTENTS ****

```
      NTIME = 0                                                   90021590
C                                                                 90021600
C ............................................................... 90021610
C                                                                 90021620
  150 CALL ALIFE                                                  90021630
C                                                                 90021640
      IF(HCALC) 500,800,160                                       90021650
  160 IF(ABS(HCALC-HREQ)/HREQ - CON(13)) 200,200,300              90021660
  200 CONTINUE                                                    90021670
      IF(IP(57))5007,5007,992
 5007 CONTINUE
      S(1) =D(24)                                                 90021679
      S(2) = D(24)                                                90021680
      DO 210 N=1,NPT                                              90021690
      S(1) = S(1)+DAM1(N)                                         90021700
      S(2) = S(2)+DAM2(N)                                         90021710
  210 CONTINUE                                                    90021720
C  MINER'S LAW    SUM DAM1 + B*SUM DAM2 = 1/SF                    90021728
C                   B IS S(3)                                     90021729
      S(3)=(RSF - S(1))/S(2)                                      90021730
      S(5)= SF*100.                                               90021735
      S(6)=S(5)*S(3)                                              90021740
C                 TO CALCULATE PERCENT OF TOTAL FOR EACH DAMAGE   90021750
      DO 220 N=1,NPT                                              90021760
      PC1(N) = DAM1(N)*S(5)                                       90021770
      PC2(N) = DAM2(N)*S(6)                                       90021780
  220 CONTINUE                                                    90021790
C                                                                 90021800
      S(5)=S(1)*S(5)                                              90021801
      S(6)=S(2)*S(6)                                              90021802
C                                                                 90021803
      S(4) = FMAX(NTOP)/FTU*100.                                  90021810
      WRITE(6,250)NTIME,HCALC,HREQ,FMAX(NTOP),S(4)
  250 FORMAT(1H1,10X,43HNUMBER OF ITERATIONS IN SUBROUTINE FATIGU =,I4,
     1   31X,21H** FATIGU - IP(57) **/
     *            /17X, 14HCALC. LIFE(HR), 9X 17HREQUIRED LIFE(HR), /
     * 16X,1E14.6,10X,1E14.6,6X,13HHIGHEST FMAX=,1E13.6,2H =,1F7.2,
     * 7HPCT FTU)
      WRITE(6,255)(N,FMAX(N),FMIN(N),CYC(N),DAM1(N),PC1(N),DAM2(N),   90021860
     * PC2(N), N=1,NPT )                                              90021870
  255 FORMAT(//8X,1HN, 7X, 4HFMAX, 8X, 4HFMIN, 5X, 8HAPP.CYC., 7X,   90021880
     * 8HDAMAGE 1,7X,3HPCT, 7X, 8HDAMAGE 2, 7X,3HPCT// (I19, 3F12.0,  90021890
     * 1E17.6, 1F8.2, 1E17.6, 1F8.2) )                                90021900
      WRITE(6,257) S(1),S(5),S(2),S(6),S(3)                           90021901
  257 FORMAT(// 13X,32HCUMULATIVE DAMAGE FOR EACH BLOCK, 2(E17.6,F8.2) 90021902
     *  //8X, 32HNUMBER OF RESIDUAL DAMAGE BLOCKS,1F9.3)              90021903
      GO TO 999                                                   90021910
C                                                                 90021920
  300 NTIME = NTIME + 1                                           90021930
      IF(NTIME-500)301,800,800
  301 IF(NTIME - 2) 305,350,400                                   90021950
C FIRST TIME ITTERATION   USE POINT-SLOPE  LOG-LOG LINE LIFE VS. SUMFMAX90021960
C     WHERE  CON(14) IS AVERAGE SLOPE OF DATA EXAMINED.           90021970
  305 SF1 = ALOG(SUMF)                                            90021980
      SFM = SF1                                                   90021990
C                  SFM IS PRIOR POINT FOR 2ND TIME AND AFTER      90022000
      H1 = ALOG(HCALC)                                            90022010
      S(1) = CON(14)*(HR-H1)                                      90022020
      SFN = S(1)+SF1                                              90022030
      FACT = EXP(S(1))                                            90022040
  320 DO 325 N=1,NPT                                              90022050
      FMAX(N) = FMAX(N)*FACT                                      90022060
      FMIN(N) = FMIN(N)*FACT                                      90022070
  325 CONTINUE                                                    90022080
      TF = TF*FACT                                                90022090
      IF(IP(58))5009,5009,150
 5009 IF(NTIME - 1)5010,5010,5011
 5010 WRITE(6,5012)
 5012 FORMAT(1H1,88X,21H** FATIGU - IP(58) **)
 5011 WRITE(6,330) NTIME, HCALC, FACT, TF                         90022110
  330 FORMAT(13H0***** NTIME=,I13,7X,6HHCALC=,1E15.7,7X,7HFACTOR=,    90022120
```

```
CARD NO    ****                      CONTENTS                              ****

           * 1E15.7,7X,3HTF=,1E15.7)                                   90022130
           WRITE(6,331) (S(N),N=90,96),S(1))                           90022140
       331 FORMAT (1H , 5E18.7)                                        90022150
           GO TO 150                                                   90022160
     C                                                                 90022170
     C                         2ND TIME ITERATE USE 2 POINTS           90022180
       350 HN = ALOG(HCALC)                                            90022190
           SFN = SFN                                                   90022200
           IF(HN-H1) 365,800,360                                       90022210
       360 H2 = HN                                                     90022220
           SF2= SFN                                                    90022230
           GO TO 370                                                   90022240
       365 H2 = H1                                                     90022250
           SF2= SF1                                                    90022260
       367 H1 = HN                                                     90022270
           SF1= SFN                                                    90022280
       370 SFN = (SF2-SF1)/(H2-H1)*(HR-H1)+SF1                         90022290
           FACT = EXP(SFN-SFN)                                         90022300
     C                         FACTOR IS VALUE SFN/SFN SO EXP(L(   )-LN(SFN))  90022310
     C                           ( NEXT ESTIMATED Y OVER PR   IOUS Y )  90022320
           GO TO 320                                                   90022330
     C                                                                 90022340
     C                  3RD TIME AND AFTER ASSUME HN WILL GO IN RIGHT DIRECTION 90022350
     C                         SINCE CURVES ARE SINGLE VALUED,NO INFLECTIONS.   90022360
       400 HN = ALOG(HCALC)                                            90022370
           SFN = SFN                                                   90022380
           IF(HN-H1) 365,800,405                                       90022390
       405 IF(HN-H2) 410,800,420                                       90022400
       410 IF(HN-HR) 367,800,360                                       90022410
       420 H1 = H2                                                     90022420
           SF1=SF2                                                     90022421
           GO TO 360                                                   90022422
     C                                                                 90022423
     C  IF HCALC IS NEGATIVE FIRST TIME, TRY LOWER VALUES 5 TIMES      90022430
       500 IF(NTIME .NE. 0) GO TO 800                                  90022432
           IF(MTIME .LT. 5) GO TO 510                                  90022434
           WRITE(6,505)                                                90022436
       505 FORMAT( 34H0 NEGATIVE LIFE THRU 5 ITTERATIONS )             90022438
           GO TO 800                                                   90022440
       510 MTIME = MTIME + 1                                           90022442
           FACT = D(19)                                                90022444
           GO TO 320                                                   90022446
     C                                                                 90022449
     C ******* ERROR PRINT .... HCALC = 0.0 ROOT CALC IN ALIFE ERROR   90022450
     C                         HCALC = SOME VALUE - FAILED TO REACH    90022460
     C                               HREQ IN FATIGU BUT CANT CLOSE IN. 90022461
     C                                                                 90022462
       800 GO TO (801,803,805,807),K                                   90022465
       801 WRITE(6,802)                                                90022466
       802 FORMAT( 38H0WORKING ON WING AT SIDE OF FUSELAGE )           90022467
           GO TO 809                                                   90022468
       803 WRITE(6,804)                                                90022469
       804 FORMAT( 38H0WORKING ON WING OUTER PANEL STATION )           90022470
           GO TO 809                                                   90022471
       805 WRITE(6,806)                                                90022472
       806 FORMAT( 38H0WORKING OF FUSELAGE PRESSURE CYCLES )           90022473
           GO TO 809                                                   90022474
       807 WRITE(6,808)                                                90022475
       808 FORMAT( 38H0WORKING ON FUSELAGE ENDURANCE LIMIT )           90022476
       809 WRITE(6,810) NTIME,HCALC,FTU,E,RA,HREQ,ANPT,SF, ( FMAX(N),FMIN(N),90022480
           * CYC(N),N=1,NPT )                                          90022490
       810 FORMAT(48H0*** ERROR IN FATIGUE, PERTINENT DATA FOLLOWS... /  90022500
           * 7H NTIME=,I13,4X,6HHCALC=, 1E14.6,4X,4HFTU=,1E14.6,4X,2HE=,  90022510
           * 1E14.6,4X,3HRA=,1E14.6 /14X,6HHREQ =,1E14.6,4X,5HANPT=,1E14.6, 90022520
           * 4X,3HSF=,1E14.6//15X,4HFMAX,14X,4HFMIN,14X,3HCYC /(6X,3E18.6) ) 90022530
           WRITE(6,820) (N, S(N),S(N+1),S(N+2),S(N+3),S(N+4), N=1,96,5)  90022540
       820 FORMAT(1H0,16HCALCULATION AREA // (I16, 5E18.7 ))           90022550
           ND(44) = 1                                                  90022560
       999 RETURN                                                      90022570
           END                                                         90022580
     C
```

```
CARD NO  ****                         CONTENTS                              ****

C ''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
C                       SUBROUTINE FTGCTL
C ''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
C
      SUBROUTINE FTGCTL                                                 90010010
C
      COMMON TCOM(5600)                                                 90010020
      COMMON /MISC/XMISC(100)
      COMMON /IPRINT/ IP(100)
C
      DIMENSION SAVE(16)                                                90010022
      DIMENSION D(2400), T(3000), ND(200)                               90010030
     1, FMAX(300), FMIN(300), AN(300)                                   90010040
     2, BMA(3,220), EM(220), BMREF(20), NSEQ(2,10)                      90010041
     *, BMSMX(2), TMD(300)                                              90010050
     *, WIND(2), SFW(2), FKTW(2)                                        90010060
     *, CON(30)                                                         90010070
     *, FPR(300), PMX(100), PMN(100), PCY(100)                          90010080
      EQUIVALENCE (D(1),TCOM(1)), (T(1),TCOM(2401)), (ND(1),TCOM(5401)) 90010100
     *,          (D(101),M), (D(102),SF), (D(103),FKT), (D(104),FTU)    90010105
     *, (D(105),E), (D(106),RA),(D(107),ANPT)                           90010110
     *, (D(51),CON(1)), (D(81),BLOCK)                                   90010120
     *, (D(109),WIND(1)), (D(111),FIND), (D(112),FFIND)                 90010130
     *, (D(113),SFW(1)), (D(115),FKTW(1)), (D(117),SFFP), (D(118),FKTFP)90010141
     *, (D(201),FMAX(1)), (D(501),FMIN(1)), (D(801),AN(1))              90010149
     *, (D(1161),BMMX), (D(1162),BMSMX(1)), (T(1101),TMD(1))            90010150
     *, (D(1200),FPT), (D(1201),FPR(1)),        (T(41),HRBL)            90010160
     *  (FPR(1),PMX(1)), (FPR(101),PMN(1)), (FPR(201),PCY(1))           90010170
     *, (D(1501),BMA(1,1)), (D(2127),EM(1)), (D(2107),BMREF(1))         90010180
     *, (ND(40),NPT)                                                    90010190
     *,  (ND(121),NSEQ(1,1)), (ND(42),NTOP), (ND(44),IFERR)             90010200
     *, (ND(45),MI), (ND(47),K)                                         90010210
     *, (ND(93),IF3), (ND(59),NMATL)                                    90010220
C TO BE OBTAINED FROM MATERIAL LIBRARY ------------------------------- 90010230
C     RA  GENERAL TMD(5)  REDUCTION AREA                                90010240
C     FTU FROM 1ST TEMPERATURE  TMD(126)                                90010250
C     FTPL FROM 1ST TEMP. TMD(121)                                      90010260
C    ETPL (EPSILON TEMP. LIMIT)  1ST TEMP  TMD(119)                     90010270
C     E =  FTPL/ETPL                                                    90010280
C TO BE SAVED IN MATERIAL LIBRARY                                       90010290
C     KFTU FACTOR FOR ULT FATIGUE EFFECT.                               90010300
C         FOR FUSALAGE TMD(132) 1ST TEMP.(SAME REL.LOC. EACH TEMP (23))90010310
C         FOR WING SIDE OF FUS. TMD(133) .... TEMP(24)                  90010320
C         FOR WING OUTBOARD (WOS)  TMD(134) ... TEMP(25)                90010330
C         FOR ENDURANCE LIMIT TMD(130) ..  TEMP(21)                     90010331
C        -----------------------------------------------------          90010332
C
      DO 10 N=1,16                                                      90010335
   10 SAVE(N) = 0.0                                                     90010336
C
C    WIND(1) WING INDICATOR FOR SIDE OF FUSELAGE, VALUE IS MATERIAL NO.90010340
C    WIND(2) WING INDICATOR FOR OUTBOARD STA.  VALUE IS MATERIAL NO.   90010350
C    FIND FUSELAGE COVER INDICATOR, VALUE IS MATERIAL NO.               90010360
C    FFIND FUSELAGE MINOR FRAME INDICATOR, VALUE IS MATERIAL NO.        90010365
C                                                                       90010370
C    IF MATERIAL NO. IS INPUT IN FATIGUE DATA AS APOSITIVE VALUE        90010371
C          THAT MATERIAL WILL BE USED  NOT ONE FROM WEIGHT MODULE       90010372
C    IF MATERIAL NO. IS INPUT IN FATIGUE DATA AS A NEGATIVE VALUE       90010373
C          NO CALCULATION WILL BE DONE FOR THAT ITEM                    90010374
C    IF MATERIAL NO. IN THE FATIGUE DATA IS ZERO (OR NOT INPUT)         90010375
C          IT IS SETUP FROM THE CORRESPONDING WEIGHT MODULE DATA.       90010376
C                                                                       90001377
      IF(WIND(1)) 100,101,105                                           90010380
  100 WIND(1)=D(24)                                                     90010382
      GO TO 105                                                         90010384
  101 WIND(1)=XMISC(15)                                                 90010386
  105 IF(WIND(2)) 106,107,110                                           90010388
  106 WIND(2)=D(24)                                                     90010390
      GO TO 110                                                         90010392
  107 WIND(2)=XMISC(15)                                                 90010394
  110 IF(FIND) 111,112,113                                              90010396
```

11/02/73 INPUT LISTING AUTOFLOW CHART SET - SWEEP FATIGUE MODULE

CARD NO **** CONTENTS ****

```
  111 FIND=D(24)
      GO TO 113                                                        90010398
  112 FIND=XMISC(31)                                                   90010400
  113 IF(FFIND) 114,115,116                                            90010402
  114 FFIND=D(24)                                                      90010404
      GO TO 116                                                        90010410
  115 FFIND=XMISC(41)                                                  90010420
  116 CONTINUE                                                         90010430
      HRBL=H*BLOCK                                                     90010440
C                                                                      90010450
C                                                                      90010450
  200 IF(WIND(1) .EQ. D(24)) GO TO 266                                 90010460
C FOR BENDING MOMENTS AS STRESS LEVELS...CONSIDER ONLY WHEN THE        90010470
C MAXIMUM (1ST HALF OF EACH SPECTRA SEGMENT) IS POSITIVE.              90010480
C BECAUSE CRITICAL CONDITION IS  COMPRESSION ON UPPER SURFACE.         90010490
C      TENSION ON LOWER SURFACE.                                       90010500
C                                                                      90010510
C BENDING MOMENTS ARE AVAILABLE FOR (1) SIDE OF FUSELAGE AND           90010520
C    (2) ONE OUTBOARD STATION. MOMENTS  FOR SWEPT COND. IF APPLICABLE. 90010530
C     B.M. ARE IN 8 FLIGHT SEGMENTS  + 2 TAXI SEG. +  1 GROUND-AIR-GR. 90010540
C     FIRST 10 HAVE 20 PTS (10 MAX., 10MIN.) LAST HAS 2 PTS.           90010550
C     BMA(K,L)    L=POINT COUNTER                                      90010560
C     K =1 SOF, SIDE OF FUS.                                           90010570
C     K =2 WOS  2ND WING STA.                                          90010580
C     K =3 EXCEEDANCES FOR GUST,                                       90010590
C    EM =  EXCEEDANCES FOR MANEUVER .                                  90010600
C EXCEEDANCES FOR MANEUVER ARE CALC ONLY 1ST 8 BUT USE 10 TO SIMPLIFY. 90010610
C ---------------------------------------------------------------------90010611
      K=1                                                              90010620
C*** READ MATERIAL LIBRARY(WITH MATERIAL NUMBER SET= WIND(K) ) INTO TMD 90010630
  201 IF3 = WIND(K)                                                    90010640
      IF(IF3) 2012,2012,2011                                           90010650
 2011 IF(IF3-NMATL) 2015,2015,2012                                     90010660
 2012 IF3 = 1                                                          90010670
      WRITE (6,2013) WIND(K), NMATL                                    90010680
 2013 FORMAT(///1HO, 18H***** MATERIAL NO., 1F5.0, 7H BEYOND, 114,     90010690
     *18H, USED MATL. NO. 1  )                                         90010700
C                                                                      90010710
 2015 IF3 = IF3 + 40
      CALL READMS(1,TMD(1),300,IF3)
      FTU=TMD(128)                                                     90010730
      RA =TMD(5)                                                       90010740
      E  =TMD(121)/TMD(119)                                            90010750
C                                                                      90010760
C     SET MI, MATERIAL INDICATOR = 0 = 1ST OR NEW MATERIAL             90010770
      MI = 0                                                           90010780
  202 N=0                                                              90010790
C
      IND = (K-1) * 3
      SAVE(IND+1) = IF3 - 40
      SAVE(IND+2) = TMD(K+132)
C
      DO 212 LL=1,10                                                   90010800
      L=(LL-1)*20+1                                                    90010810
      M=L+10                                                           90010820
      IREF=LL+(K-1)*10                                                 90010821
      DO 210  J=1,9                                                    90010830
      I=J-1                                                            90010840
C              FIRST DO GUST                                           90010838
      T(1) = (BMA(K,L+I)-BMA(K,L+J))/D(3) + BMA(K,L+J)                 90010850
      IF(T(1) .LE. D(24)) GO TO 210                                    90010860
      N=N+1                                                            90010870
      T(2) = (BMA(K,M-I)-BMA(K,M-J))/D(3) + BMA(K,M-J)                 90010875
      FMAX(N) = T(1)                                                   90010880
      FMIN(N) = T(2)                                                   90010890
      AN(N) = BMA(3,L+J) - BMA(3,L+I)                                  90010891
C     FOR MANEUVER BENDING MOMENTS ARE THE SAME BUT EXCEEDANCES IN EM. 90010892
      T(3) = EM(L+J) - EM(L+I)                                         90010893
      IF(T(3) .LE. 0.0) GO TO 206                                      90010894
      IF( T(1) - ZF(IREF) ) 203,206,204                                90010895
  203 N=N+1                                                            90010896
```

```
CARD NO  ****                    CONTENTS                                 ****

              FMAX(N) = BMREF(IREF)                                    90010897
              FMIN(N) = T(1)                                           90010898
              GO TO 205                                                90010899
         204  N=N+1                                                    90010900
              FMAX(N) = T(1)                                           90010901
              FMIN(N) = BMREF(IREF)                                    90010902
         205  AN(N) = T(3)                                             90010903
         206  T(4) = EM(M-J) - EM(M-1)                                 90010904
              IF(T(4) .LE. 0.0) GO TO 210                              90010905
              IF(T(2) - BMREF(IREF) ) 207,210,208                      90010906
         207  N=N+1                                                    90010907
              FMAX(N) = BMREF(IREF)                                    90010908
              FMIN(N) = T(2)                                           90010909
              GO TO 209                                                90010910
         208  N=N+1                                                    90010911
              FMAX(N) = T(2)                                           90010912
              FMIN(N) = BMREF(IREF)                                    90010913
         209  AN(N) = T(4)                                             90010914
         210  CONTINUE                                                 90010915
C                                                                      90010916
               NSEQ(K,LL)=N                                            90010920
         212  CONTINUE                                                 90010930
              N=N+1                                                    90010940
              FMAX(N) = BMA(K,201)                                     90010950
              FMIN(N) = BMA(K,202)                                     90010960
              AN(N) = (BMA(3,201)+BMA(3,202))/D(2)                     90010970
              ANPT = N                                                 90010980
              NPT = N                                                  90010990
              SF = SFW(K)                                              90010991
              FKT = FKTW(K)                                            90010992
              IF(IP(57))5021,5021,5002
         5021 CONTINUE
              WRITE(6,80) H,SF,FKT,FTU,E,RA,NPT                        90011000
           80 FORMAT(1H1,20HFATIGUE  INPUT DATA ,68X,21H** FTGCTL - IP(57) **/
             1                                /1X, 2HH=,1E14.7, 5X, 3HSF=,
             *1E14.7, 5X, 3HKT=,1E14.7//1X,4HFTU=,1E14.7, 5X, 2HE=,1E14.7, 5X,
             *3HRA=, 1E14.7, 20X, 4HNPT=,I14  ///)
              IF(K.EQ.2) GO TO 215                                     90011010
              WRITE(6,214)                                             90011020
         214  FORMAT(17HOSIDE OF FUSELAGE )                            90011030
              GO TO 220                                                90011040
         215  WRITE(6,216)                                             90011050
         216  FORMAT(22HOWING OUTBOARD STATION )                       90011060
         220  CONTINUE                                                 90011070
C BMMX = MAX FROM SPECTRA FOUND HERE                                    90011080
         5002 CONTINUE
C BMSMX=  STATIC MAX FOUND ELSEWHERE AND PASSED (2 VALUES) ------------ 90011090
              BMMX = D(24)                                             90011100
              DO 222 N=1,NPT                                           90011110
              IF(FMAX(N) .GT. BMMX) BMMX=FMAX(N)                       90011120
         222  CONTINUE                                                 90011130
C      MODIFY BM TO SIMULATE STRESS. . . . . . . . . . . . . . . . .   90011140
              T(1) = CON(28)*FTU/BMMX                                  90011150
              DO 225 N=1,NPT                                           90011160
              FMAX(N) = FMAX(N)*T(1)                                   90011170
         225  FMIN(N) = FMIN(N)*T(1)                                   90011180
              IF(IP(57))5003,5003,5004
         5003 WRITE(6,230) T(1)                                        90011190
         230  FORMAT( 42HOSTRESS LEVELS SET UP FROM BENDING MOMENTS,   90011200
             *, 2X, 5HTIMES, 1E13.6 ////)                              90011210
              WRITE(6, 85) (I,FMAX(I),FMIN(I),AN(I), I=1,NPT)          90011220
           85 FORMAT(I110, 3E19.7)                                     90011230
         5004 CONTINUE
C                                                                      90011240
              CALL FATIGU                                              90011250
C                                                                      90011260
              IF(IFERR) 240,250,240                                    90011270
         240  SAVE(INO+3)=SAVE(INO+2)                                  90011280
              GO TO 260                                                90011290
         250  T(1) = FMAX(NTOP)/FTU * BMSMX(K)/BMMX                    90011300
         251  DO 255 I=132,257,25                                      90011310
```

```
           THD(I+K) = T(I)
       255 CONTINUE                                                   90011320
           SAVE(IND+3) = T(I)                                         90011330
       260 IF(IP(57))5007,5007,5008
      5007 WRITE(6,265) (I,NSEQ(K,I),I=1,10)
       265 FORMAT(1H1,10X,41HN FOR END OF EACH SPECTRA SEGMENT FOLLOWS ,
          137X,21H** FTOCTL - IP(57) **// (11X,I13,I10))              90011350
      5008 CONTINUE
           CALL WRITMS(1,THD(1),300,IF3)
           IF(K .EQ. 2) GO TO 500
       266 IF(WIND(2) .EQ. D(24)) GO TO 500                           90011390
           K=2                                                        90011400
           IF(WIND(1).NE.WIND(2)) GO TO 201                           90011410
           MI = 1                                                     90011420
           GO TO 202                                                  90011430
     C                                                                90011440
     C ..............................................................90011450
     C  FUSELAGE PRESSURES TO SET UP STRESS LEVELS                    90011460
     C                                                                90011470
       500 IF(FIND .EQ. D(24)) GO TO 800                              90011480
           KT=0                                                       90011490
           IF(WIND(1)+WIND(2).EQ.D(24))  GO TO 5001                   90011491
           IF(FIND .EQ. WIND(K)) GO TO 507                            90011500
     C*** READ MATERIAL LIBRARY WITH MATERIAL NO. SET= FIND           90011510
      5001 IF3=FIND                                                   90011520
           IF(IF3)502,502,501                                         90011530
       501 IF(IF3-NMATL) 503,503,502                                  90011540
       502 IF3=1                                                      90011550
           WRITE(6,2013) FIND, NMATL                                  90011560
       503 IF3 = IF3 + 40                                             90011570
           CALL READMS(1,THD(1),300,IF3)
           RA =THD(5)
           FTU=THD(26)                                                90011590
           E =THD(21)/THD(19)                                         90011600
           MI = 0                                                     90011610
           GO TO 510                                                  90011620
       507 MI = 1                                                     90011630
     C FPT IS NUMBER OF FUSELAGE N/J. PRESSURES                       90011640
       510 NPT = FPT                                                  90011650
     C                                            SAVE FOR PRINT AT END 90011660
           SAVE(KT+7)=IF3-40                                          90011661
           SAVE(KT+8)=THD(130)                                        90011662
           SAVE(KT+10)=THD(132)                                       90011663
     C                                                                90011664
           ANPT = FPT                                                 90011665
           SF = SFTP                                                  90011670
           FKT = FKTFP                                                90011671
           IF(IP(57))5005,5005,5006                                   90011672
      5005 WRITE(6,801) H,SF,FKT,FTU,E,RA,NPT
           IF(KT .EQ. 5) GO TO 5052                                   90011680
           WRITE(6,5051)                                              90011681
      5051 FORMAT(/ 15H0FUSELAGE COVER )                              90011682
           GO TO 5006                                                 90011683
      5052 WRITE(6,5053)                                              90011684
      5053 FORMAT(/ 22H0FUSELAGE MINOR FRAMES)                        90011685
      5006 CONTINUE                                                   90011686
     C ASSUMED PRESSURE BLOCK HAS ARRANGED IN SEPERATE ARRAYS         90011687
     C      WHERE FPR(1) TO (100)  ARE  PMAX                          90011690
     C            FPR(101) TO (200) ARE PMIN                          90011700
     C            FPR(201) TO (300) ARE APPLIED CYCLES                90011710
           T(1) = D(24)                                               90011720
           DO 520 N=1,NPT                                             90011730
           FMAX(N) = PMX(N)                                           90011740
           FMIN(N) = PMN(N)                                           90011750
           AN(N)   = PCY(N)                                           90011760
           IF(FMAX(N).GT. T(1)) T(1)=FMAX(N)                          90011770
       520 CONTINUE                                                   90011780
           T(2) = CON(28)*FTU/T(1)/D(2)                               90011790
           DO 525 N=1,NPT                                             90011800
           FMAX(N) = FMAX(N)*T(2)                                     90011810
           FMIN(N) = FMIN(N)*T(2)                                     90011820
                                                                      90011830
```

CARD NO **** CONTENTS ****

```
      525  CONTINUE                                                          90011840
           IF(IP(57))5009,5009,5010
      5009 WRITE(6,530) T(2)                                                 90011850
      530  FORMAT(52H0STRESS LEVELS SET UP FROM FUSELAGE PRESSURES TIMES ,   90011860
          * 1E13.6 /// )                                                     90011870
           WRITE(6,85) (N, FMAX(N), FMIN(N), AN(N), N=1,NPT )                90011880
      5010 CONTINUE
           K=3                                                               90011889
           CALL FATIGU                                                       90011890
           IF(IFERR)590,600,590                                              90011900
      590  SAVE(KF+11)=SAVE(KF+10)                                           90011910
           GO TO 617                                                         90011920
      600  T(1) = FMAX(NTOP)/FTU                                             90011930
      610  DO 615 I=132,257,25                                               90011940
           THD(I) = T(1)                                                     90011950
      615  CONTINUE                                                          90011960
           SAVE(KF+11)=T(1)                                                  90011965
     C SET-UP FOR ENDURANCE LIMIT, MATERIAL SAME AS FOR FUS. PRESSURE        90011970
      617  MI=1                                                              90011980
           ANPT = D(1)                                                       90011990
           NPT = 1                                                           90012000
           FMAX(1)=FTU/D(2)                                                  90012010
           FMIN(1)=-FMAX(1)                                                  90012020
           AN(1) = CON(29)                                                   90012030
           FKT = CON(30)                                                     90012031
           SF = D(1)                                                         90012032
           IF(IP(57))5011,5011,5012
      5011 WRITE(6,620)                                                      90012040
      620  FORMAT(/// 16H0ENDURANCE LIMIT  )                                 90012050
           WRITE(6,85)  NPT, FMAX(1), FMIN(1), AN(1)                         90012060
      5012 CONTINUE
           K=4                                                               90012069
           CALL FATIGU                                                       90012070
           IF(IFERR) 650,660,650                                             90012080
      650  SAVE(KF+9)=SAVE(KF+8)                                             90012090
           GO TO 675                                                         90012100
      660  T(1) = FMAX(1)/FTU                                                90012110
      665 DO 670 N=130,255,25                                                90012120
           THD(N) = T(1)                                                     90012130
      670  CONTINUE                                                          90012140
           SAVE(KF+9)=T(1)                                                   90012150
      675  CONTINUE                                                          90012160
           CALL WRITMS(1,THD(1),300,IF3)
     C
           IF(KF .NE. 0) GO TO 999                                           90012190
      800  IF(FFIND .EQ  D(24)) GO TO 999                                    90012200
           IF(FFIND .EQ. FIND) GO TO 820                                     90012210
           KF=5                                                              90012220
           IF3=FFIND                                                         90012230
           IF(IF3) 802,802,801                                               90012240
      801  IF(IF3-NMATL) 503,503,802                                         90012250
      802  IF3=1                                                             90012260
           WRITE(6,2013) FFIND, NMATL                                        90012270
           GO TO 503                                                         90012280
     C                           IF MINOR FRAME AND COVER SAME MATERIAL      90012290
      820  DO 822 N=7,11                                                     90012300
           SAVE(N+5)=SAVE(N)                                                 90012310
      822  CONTINUE                                                          90012320
     C                                                                       90012330
      999  CONTINUE                                                          90012340
     C                                                                       90012350
      5013 WRITE(6,998) (SAVE(N),N=1,18)                                     90012360
      998 FORMAT(1H1,54HCHANGES MADE TO MATERIAL PROPERTIES BY FATIGUE PROGR
          *AM,43X,12H** FTGCTL **/
          1   / 15X,27HSIDE OF FUSELAGE ** MATL NO,F4.0/
          *  11X,21HTHD(133) CHANGED FROM,F8.4,4H TO ,F8.4//
          *     18X,25HWING STATION 2 ** MATL NO,F4.0/
          *  11X,21HTHD(134) CHANGED FROM,F8.4,4H TO ,F8.4//
          *     18X,25HFUSELAGE COVER ** MATL NO,F4.0/                       90012430
          *  11X,21HTHD(130) CHANGED FROM,F8.4,4H TO ,F8.4/                  90012440
          *  11X,21HTHD(132) CHANGED FROM,F8.4,4H TO ,F8.4//                 90012450
```

```
11/02/73      INPUT LISTING                       AUTOFLOW CHART SET - SHEEP    FATIGUE MODULE

 CARD NO      ****                    CONTENTS                                      ****

   994              *     13X,31HFUSELAGE MINOR FRAME ** MATL NO,F4.0/           80012960
   995              *  11X,21HTMD(130) CHANGED FROM,F8.4,4H TO ,F8.4/             80012980
   996              *  11X,21HTMD(132) CHANGED FROM,F8.4,4H TO ,F8.4)             80012990
   997        C                                                                  80012970
   998         5014 RETURN                                                       80012980
   999              END                                                          80012990
```

Section V

## REFERENCES


1. Manson, S.S., "Fatigue: A Complex Subject - Some Simple Approximations," Experimental Mechanics, July 1965

2. Manson, S.S., Thermal Stress and Low Cycle Fatigue, McGraw-Hill, 1966

3. Petersen, R.E., "Fatigue of Metals," Part 3, ASTM Materials Research and Standards, February 1963, p 122-139

4. Stowell, E.Z., "Stresses and Strain Concentrations at a Circular Hole in an Infinite Plate,"NACA TN2073, 1950

5. Tetelman and McErily, "Fatigue Tests on Notched and Unnotched Sheet Specimens of 2024-T3 and 7075-T6 Aluminum Alloys and of SAE 4130 Steel With Special Consideration of the Life Range From 2 to 10,000 Cycles," NACA TN 3866, 1956

6. Benham, P.P., and Ford, H., "Low Endurance Fatigue of a Mild Steel and an Aluminum Alloy," Journal Mechanical Engineering Science, Institute Mechanical Engineers, June 1961

7. Coffin, L.F., "The Problem of Thermal Stress Fatigue in Austenitic Steels at Elevated Temperatures," ASTM STP No. 165, 1954, p 31

8. Kuhn, P., and Hardrath, H.F., "An Engineering Method for Estimating Notch Size Effect in Fatigue Tests on Steel," NACA TN 2805, 1952

9. Petersen, R.E., Stress Concentration Design Factors, John Wiley & Sons, 1953

10. Bowman, C.E., and Dolan, T.J., "Biaxial Fatigue Properties of Pressure Vessel Steels,"Welding Journal Research Supplement, November 1953

11. Tejani, S., "A Computer Program for Fatigue Life Prediction Based on Strain,"B-1 Division/Rockwell International, NA-72-239, 16 July 1973